



INSTITUTO FEDERAL

São Paulo

Campus Campinas

LUANA MATALLO RUGGIERO

**Design de Interface Front-End de uma aplicação web para organização
pessoal de leitura de livros**

CAMPINAS
2024

Luana Matallo Ruggiero

Design de Interface Front-End de uma aplicação web para organização
pessoal de leitura de livros

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para obtenção do
diploma do Curso Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto
Federal de Educação, Ciência e Tecnologia
Campus Campinas.

Orientador: Prof. Dra. Bianca Maria Pedrosa

CAMPINAS

2024

FICHA CATALOGRÁFICA

ATA N.º 19/2024 - TADS-CMP/DAE-CMP/DRG/CMP/IFSP

Ata de Defesa de Trabalho de Conclusão de Curso - Especialização

Na presente data, realizou-se a sessão pública de defesa do Trabalho de Conclusão de Curso intitulado "Design de Interface Front-End de uma aplicação web para organização pessoal de leitura de livros", apresentado(a) pelo(a) aluno(a) Luana Matallo Ruggiero do **CURSO DE TECNÓLOGO EM ANÁLISE DE SISTEMAS** (campus Campinas). Os trabalhos foram iniciados às 15:00 h pelo(a) Professor(a) presidente da banca examinadora, constituída pelos seguintes membros:

Membros	Instituição	Presença (Sim/Não)
Bianca Maria Pedrosa (Presidente/Orientador)	IFSP	sim
Eliana Alves Moreira (avaliador 1)	IFSP	sim
Antônio Queiroz da Silva Neto (avaliador 2)	IFSP	sim

Observações:

A banca examinadora, tendo terminado a apresentação do conteúdo da monografia, passou à arguição do(a) candidato(a). Em seguida, os examinadores reuniram-se para avaliação e deram o parecer final sobre o trabalho apresentado pelo(a) aluno(a), tendo sido atribuído o seguinte resultado:

☒ Aprovado(a) ☐ Reprovado(a)

Proclamados os resultados pelo presidente da banca examinadora, foram encerrados os trabalhos e, para constar, eu lavrei a presente ata que assino em nome dos demais membros da banca examinadora.

IFSP Campus Campinas - 14/12/2023

Documento assinado eletronicamente por:

- Bianca Maria Pedrosa, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 05/12/2024 15:52:00.
- Eliana Alves Moreira, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 05/12/2024 15:53:00.
- Antonio Queiroz da Silva Neto, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 05/12/2024 15:53:46.

Este documento foi emitido pelo SUAP em 05/12/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifsp.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 856248
Código de Autenticação: 5b917809c2



*Dedico este trabalho a minha família,
professores e grandes amigas, mas principalmente
para minha mãe que me apoiou desde sempre.*

*"Uma biblioteca é uma ilha
num grande mar de ignorância".*

Lemony Snicket

RESUMO

O projeto aborda o desenvolvimento de uma aplicação *web* para gestão pessoal de leituras, focando em funcionalidades que aprimoram a experiência de leitura dos usuários. Com o aumento da popularidade de livros impulsionado pela comunidade BookTok, a demanda por ferramentas para organização e acompanhamento de leitura tem crescido. A aplicação proposta é projetada com React e incorpora elementos de UX/UI, permitindo aos usuários cadastrarem, organizar e acompanhar suas leituras de forma eficiente e intuitiva. O projeto baseou-se na análise de aplicações existentes como Skoob, Goodreads e Maratona, visando superar as limitações identificadas nessas plataformas, especialmente em termos de intuitividade e personalização. A aplicação oferece funcionalidades de metas de leitura, atualização de progresso, personalização de layout, e inclui um sistema de busca eficiente.

Palavras-chave: gestão de leitura; aplicação web; personalização de layout; design interface; React.

ABSTRACT

The project addresses the development of a web application for personal reading management, focusing on features that enhance the user reading experience. With the growing popularity of books driven by the BookTok community, the demand for tools to organize and track reading has increased. The proposed application is designed with React and incorporates UX/UI elements, allowing users to register, organize, and monitor their readings efficiently and intuitively. The project is based on existing applications such as Skoob, Goodreads, and Maratona, aiming to overcome the limitations identified in those platforms, especially regarding intuitiveness and customization. The application offers features such as reading goals, progress tracking, layout customization, and includes an efficient search system.

Keywords: reading management; web application; layout customization; interface design; React.

LISTA DE FIGURAS

Figura 1 - Modelo de prototipação.....	20
Figura 2 – Exemplo de Código para adicionar livros.....	24
Figura 3 - Página Inicial Skoob.....	29
Figura 4 - Página de Livro Skoob.....	31
Figura 5 - Página de Livro Skoob.....	31
Figura 6 - Página de Livros Skoob.....	33
Figura 7 - Página Inicial Goodreads.....	34
Figura 8 - Página de Livro Goodreads.....	36
Figura 9 - Página de Livro Goodreads.....	37
Figura 10 - Página de Livros Goodreads.....	38
Figura 11 - Página Inicial Maratona.....	39
Figura 12 - Página de Livro Maratona.....	40
Figura 13 - Página de Livro Maratona.....	41
Figura 14 - Página de Livro Maratona.....	41
Figura 15 - Página de Livros Maratona.....	42
Figura 16 - Paleta de cores.....	45
Figura 17 - Protótipo tela.....	46
Figura 18 - Rascunho de baixa fidelidade.....	48
Figura 19 - Prototipação de Alta fidelidade.....	49
Figura 20 - Protótipo de Alta fidelidade – Fluxo.....	49
Figura 21 - Protótipo de Alta fidelidade - Tela Inicial.....	50
Figura 22- Protótipo de Alta fidelidade - Cadastro Livro.....	51
Figura 23 - Protótipo de Alta fidelidade - Tela Configuração.....	52
Figura 24 – useState.....	54
Figura 25 – useEffect.....	55
Figura 26 - Adicionar e Gerenciar Livros.....	56
Figura 27 – Renderização.....	57
Figura 28 - Dados em Book.....	57
Figura 29 - Progresso em Book.....	58
Figura 30 – BookGallery.....	58
Figura 31 – BookList.....	59
Figura 32 - Salvando meta.....	60
Figura 33 - Cadastro de livro.....	61
Figura 34 - Atualização do livro.....	62
Figura 35 - Configurações do layout.....	63
Figura 36 - Cadastro de metas.....	64
Figura 37 - Meta atualizada.....	65
Figura 38 – Questionário.....	66
Figura 39 – Questionário.....	67

LISTA DE QUADROS

Quadro 1 - Análise de sites.....	42
Quadro 2 - Requisitos funcionais.....	44
Quadro 3 - Requisitos não funcionais.....	44

LISTA DE SIGLAS

APP	Application
CSS	Cascading Style Sheets
DOM	Document Object Model
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
ISBN	International Standard Book Number
JSON	JavaScript Object Notation
JSX	JavaScript XML
NoSQL	Not Only Structured Query Language
SNEL	Sindicato Nacional dos Editores de Livros
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience

SUMÁRIO

1 INTRODUÇÃO	12
2 JUSTIFICATIVA	14
3 OBJETIVOS	15
3.1 Objetivo Geral	16
3.2 Objetivos Específicos	16
4 FUNDAMENTAÇÃO TEÓRICA	17
4.2 Prototipação	17
4.3 Desenvolvimento Front-end	19
4.3.1 REACTJS	20
4.3.1.1 Experiência do Usuário (UX)	23
4.3.1.2 Interface do Usuário (UI)	24
5 METODOLOGIA	25
5.1 Modelo de Desenvolvimento	25
5.2 Comunicação	25
5.2.1 SKOOB	25
5.2.2 Goodreads	30
5.2.3 Maratona	35
5.3 Plano Rápido	42
5.4 Construção do Protótipo	44
5.5 Implementação, entrega e feedback	49
5.5.1 Implementação	49
5.5.2 Código React	50
5.5.3 Aplicação	57
6 CONSIDERAÇÕES FINAIS	65
REFERÊNCIAS	66

1 INTRODUÇÃO

Atualmente a leitura vem ganhando uma crescente popularidade, possivelmente impulsionada pelo fenômeno conhecido como Booktok, uma comunidade de usuários do TikTok com conteúdo voltados para livros e literatura em geral. O crescimento dessa comunidade, intensificado pela pandemia em 2020, trouxe um aumento exponencial na divulgação de livros e na troca de recomendações entre os usuários de forma positiva, como sugeriu o presidente do SNEL, Sindicato Nacional dos Editores de Livros, Dante Cid (Vendas, 2022).

O sentimento de pertencimento a essa comunidade virtual vem sendo um estímulo à leitura no Brasil, incentivando indivíduos, tanto aqueles que criam os conteúdos como os que os assistem e interagem, a se envolverem mais ativamente com os livros e a cultura literária (Silva; Santos, 2023). Esse movimento não só ajuda a ampliar o acesso à literatura, mas também cria uma atmosfera de compartilhamento e engajamento em torno dos livros, contribuindo para a formação de uma cultura leitora mais forte e diversificada.

Nesse cenário, a crescente demanda por organizar as leituras reflete não apenas o aumento quantitativo de livros consumidos, mas também uma busca por uma experiência mais enriquecedora e personalizada do processo de organização de leitura. A criação de listas de leitura, a avaliação e resenha de livros, assim como o acompanhamento do progresso de leitura, não apenas facilitam o gerenciamento do acervo, mas também promovem uma conexão mais pessoal e significativa com os livros, além de auxiliar no incentivo da criação de hábitos literários. A organização visa se tornar, então, uma ferramenta útil para os leitores estabelecerem e buscarem manter hábitos de leitura consistentes, além de promover um envolvimento mais ativo e reflexivo com a literatura pessoal.

Contudo, apesar da disponibilidade de aplicações populares como Skoob, Goodreads e Maratona voltadas para essa finalidade, este projeto dedica-se a desenvolver uma interface que buscar atender às demandas crescentes dos leitores contemporâneos, proporcionando não apenas recursos funcionais, mas também uma experiência de usuário agradável e acessível.

Dessa forma, o objetivo deste projeto, nesse contexto em que a demanda por uma experiência de organização de leitura mais fluida e organizada se torna cada vez

mais evidente, é desenvolver uma aplicação *web*, utilizando a tecnologia React, com o intuito de oferecer uma solução personalizável para a organização pessoal de livros. Através do desenvolvimento de uma interface cuidadosamente projetada em React, visando projetar uma experiência mais fácil e agradável ao usuário, contando com um banco de dados que vem sendo desenvolvido em um projeto correlato e que, portanto, não será abordado no decorrer deste projeto.

O intuito é que a eficácia do *design* de interface proposto suprima a necessidade dos leitores e proporcione uma experiência agradável e envolvente durante toda a jornada da organização de leitura, de modo a também contribuir significativamente para o fortalecimento da cultura leitora, promovendo a iniciativa de hábitos pessoais de leitura consistentes.

Adiante serão abordados diversos aspectos fundamentais para compreensão e desenvolvimento do projeto em questão. Inicialmente, será apresentada a justificativa e os objetivos do projeto, fornecendo uma visão das motivações por trás deste projeto e o que se espera alcançar com ele. Em seguida, será detalhada a fundamentação teórica, que servirá como base para os conceitos explorados ao longo do trabalho. Na sequência, será discutida a metodologia adotada, delineando cada etapa do processo de desenvolvimento. Por fim, serão apresentados os resultados obtidos e as conclusões derivadas desses resultados.

2 JUSTIFICATIVA

A crescente popularidade da leitura, possivelmente impulsionada pelo fenômeno do Booktok e pela ampla disponibilidade de títulos, reflete uma mudança significativa nos hábitos de consumo da literatura no Brasil (Pecinato, 2023). No entanto, esse aumento na atividade de leitura também traz consigo novas adversidades, especialmente no que diz respeito à organização e gestão das obras lidas, em andamento e futuras (Bezerra; Gandelman, 2023). Embora existam aplicações como Skoob e Goodreads, que buscam atender a essa demanda, a insatisfação dos usuários com a experiência oferecida por essas aplicações revela uma necessidade de uma nova alternativa no mercado. Desenvolvidos respectivamente entre 2006 e 2009, tanto o Skoob quanto o Goodreads apresentam uma interface desatualizada e carente de personalização, o que limita consideravelmente a eficácia da organização pessoal de leituras, dificultando o engajamento e a interação dos usuários com suas próprias bibliotecas virtuais (Bezerra; Gandelman, 2023).

No caso do Skoob, é evidente a falta de preocupação com a experiência do usuário desde sua página inicial. Sua interface é anti-intuitiva em todos os seus processos, com funcionalidades importantes frequentemente perdidas em meio a um mar de informações ou jogadas para longe da vista do usuário. Essa abordagem resulta em uma experiência insatisfatória, na qual os usuários se sentem perdidos e frustrados ao tentar encontrar o que procuram. "A interface da página inicial do perfil do usuário da Skoob, para aqueles que não têm familiaridade com ela, à primeira vista, pode ser pouco amigável, isto porque ela tem uma diversidade de informações" (Carvalho, 2021).

Por outro lado, o Goodreads demonstra um interesse maior em melhorar a experiência do usuário. No entanto, a disposição de *timeline* na página inicial, para pessoas que buscam uma experiência pessoal e não social, pode causar uma sensação de poluição visual e desorganização. Além disso, a falta de personalização da interface do Goodreads pode gerar desinteresse por parte dos usuários, que buscam uma experiência mais moderna e adaptada às suas necessidades individuais.

Esses problemas identificados com o Skoob e o Goodreads têm um impacto direto na experiência do usuário e na eficácia da organização pessoal de leituras. A falta de intuitividade e a desorganização da interface do Skoob dificultam a navegação e frustram os usuários, tornando a gestão de suas bibliotecas virtuais uma tarefa árdua. Da

mesma forma, a aparência do Goodreads pode dificultar aos usuários em busca de uma experiência mais adaptada às suas preferências individuais, reduzindo assim o engajamento e a interação com a aplicação.

Ambas as aplicações são classificadas como redes sociais devido à sua proposta de compartilhamento de leituras e interação entre os usuários. No entanto, essa semelhança nem sempre é benéfica. A natureza social dessas aplicações pode resultar em uma sobrecarga de informações e distrações, dificultando para os usuários manterem o foco em suas próprias leituras e objetivos pessoais, além da sensação de comparação e competitividade.

A relevância deste trabalho está em facilitar a organização pessoal de leitura dos usuários, tornando a experiência mais agradável e eficiente. Além disso, ao auxiliar os usuários no acompanhamento de suas metas de leitura, o projeto busca incentivar hábitos de leitura pessoais, promovendo um engajamento contínuo e uma personalização do conteúdo. Conforme destacado por Guanabara (2018) em sua pesquisa sobre a aplicação *web* skoob, os entrevistados enfatizaram que ferramentas como a 'meta de leitura' contribuem significativamente para a organização das leituras, resultando em uma maior frequência de leitura. Portanto, à medida que o usuário avança em uma leitura, ele pode se sentir mais motivado a iniciar a próxima, criando assim um hábito pessoal de leitura contínuo. Ao recomendar livros semelhantes aos que o usuário gostou, o aplicativo também espera promover a diversificação e o aumento do número de livros lidos. Ao abordar essas questões de forma abrangente e aprofundada, este projeto visa oferecer uma aplicação *web* que não apenas supere as deficiências das soluções existentes, mas também torne aos leitores a experiência de ler mais organizada.

3 OBJETIVOS

O objetivo deste projeto é desenvolver uma interface de aplicação para um gerenciador pessoal de leitura. Para alcançar esse objetivo, será implementado um controle de metas de leituras, que permitirá aos usuários estabelecerem objetivos quantitativos e acompanhar seu progresso ao longo do tempo. Será desenvolvida uma função que calcula a porcentagem de leitura de cada livro, fornecendo aos usuários uma maneira clara de acompanhar seu avanço em cada obra. Por meio de recursos como resenhas, pretende-se enriquecer as interações dos usuários com os livros, tornando a experiência de leitura mais significativa e enriquecedora.

Outro aspecto importante será possibilitar a classificação por notas e a criação de listas personalizadas, permitindo aos usuários organizarem seus livros de acordo com seus interesses e preferências individuais. Além disso, funções para personalização de *layout*, onde os usuários poderão alterar o tamanho de fontes e cores na interface. Essa personalização não apenas visa tornar o ambiente mais pessoal e limpo, mas também garantir acessibilidade, promovendo uma experiência inclusiva para todos os usuários.

Para aprimorar a funcionalidade da aplicação, será implementado um sistema de busca eficiente, permitindo que os usuários localizem rapidamente os livros em sua biblioteca. Os livros serão cadastrados pelo próprio usuário, proporcionando uma organização exclusiva e personalizada. Isso permitirá aos usuários adicionar todas as informações necessárias, garantindo que a biblioteca reflita suas preferências e necessidades individuais.

3.1 Objetivo Geral

O objetivo geral é desenvolver uma aplicação que permita aos leitores organizarem suas leituras atuais, antigas e futuras, otimizando sua rotina de leitura e proporcionando uma experiência de usuário agradável e acessível.

3.2 Objetivos Específicos

- a) Comparar as interfaces de aplicações existentes e entender seus problemas, focando na experiência do usuário, funcionalidades e acessibilidades.
- b) Desenvolver o *design* de interface focado em UX/UI, garantindo uma experiência intuitiva, agradável e acessível para os usuários.

- c) Desenvolver as funcionalidades requeridas pelos leitores, tais como, sistema de metas de leituras, cálculo de porcentagem, criação de *layout* personalizado e um sistema de busca eficiente para facilitar a localização de livros na biblioteca do usuário, entre outras.
- d) Aplicar tecnologias e frameworks mais recentes de desenvolvimento, a fim de entregar uma aplicação atualizada em termos tecnológicos.

4 FUNDAMENTAÇÃO TEÓRICA

Essa seção apresentará uma base conceitual teórica e essencial para a compreensão do trabalho desenvolvido. Aqui, serão detalhados e discutidos os conceitos fundamentais que permeiam os termos e assuntos abordados ao longo do presente trabalho, proporcionando ao leitor uma compreensão clara e profunda dos temas explorados.

Primeiramente, serão apresentados os conceitos de leitura e seus hábitos, além de um breve relato sobre o BookTok e seu incentivo à leitura. Em seguida, será demonstrada uma análise de aplicações correlatas. Por fim, haverá uma breve explicação sobre o desenvolvimento *front-end*, incluindo React, UX e UI.

Todos esses conceitos e teorias formarão a base necessária para entender as decisões de design e desenvolvimento tomadas ao longo do projeto.

4.2 Prototipação

A prototipação é um modelo de desenvolvimento muito adequado para o desenvolvimento de aplicações, pois como o nome sugere, cria-se protótipos, versões mais simples e não completamente funcionais do *software*. Assim, esses protótipos ajudam a fornecer uma visão geral sobre o *layout* e a interação da aplicação, além de permitir a análise do fluxo antes da implementação. Os protótipos são divididos em baixa fidelidade e alta fidelidade, podendo ser esboços simples até construções complexas mais detalhadas. A figura 1 apresenta as etapas de um modelo de desenvolvimento de prototipação.

Figura 1 - Modelo de prototipação.



Fonte: Pressman (2010)

As etapas da prototipação incluem:

1. **Comunicação:** Levantamento de requisitos com base na análise de trabalhos correlatos e nos objetivos definidos, serão levantados os requisitos essenciais para a interface do usuário do projeto. Isso incluirá a identificação de funcionalidades, fluxos de navegação e requisitos de design.
2. **Plano Rápido:** Definição de objetivos específicos da prototipação serão estabelecidos, incluindo a validação de conceitos de *design*, a avaliação da usabilidade. Esses objetivos guiarão o processo de criação e teste dos protótipos.
3. **Modelagem do plano rápido:** A partir da análise detalhada dos trabalhos correlatos de organização de leitura pessoal, como Skoob, Maratona e Goodreads. Serão detalhados os conceitos principais, orientando o desenvolvimento dos protótipos.
4. **Construção do Protótipo:** Serão desenvolvidos protótipos de diferentes níveis de fidelidade, desde rascunhos de baixa fidelidade até protótipos interativos de alta fidelidade. Esses protótipos serão criados utilizando ferramentas de *design* e prototipação adequadas, como o Figma.

5. Implementação:

5.1. Testes e Validação: Os protótipos serão submetidos a teste de usabilidade com usuários reais, onde serão avaliados aspectos como facilidade de uso, eficiência e satisfação do usuário.

5.2. Iteração: Com base no *feedback* recebido, os protótipos serão iterados e refinados para melhorar a experiência do usuário. Esse processo de iteração será realizado de forma contínua, garantindo que os protótipos evoluam conforme as necessidades e expectativas dos usuários.

5.3. Implementação: Por fim, quando tudo estiver adequado, o desenvolvimento será iniciado.

4.3 Desenvolvimento *Front-end*

O desenvolvimento *front-end* é baseado na parte visual e interativa de uma aplicação *web*, essencial para a criação de interfaces (W3schools, 2024c). Ele se relaciona diretamente com as partes dos sites que mais se aproximam da interação do usuário, desenvolvendo uma interface por meio de tecnologias básicas como HTML, CSS e JavaScript (Alura, 2024). Dessa forma, o *front-end* é ideal e fundamental para a UX e UI da aplicação *web*.

No contexto do cliente-servidor, o desenvolvimento *front-end* está intrinsecamente ligado às solicitações e respostas entre o navegador do cliente e o servidor que hospeda a aplicação. Por exemplo, os métodos HTTP: *Get*, *Post*, *Put* e *Delete* são comumente usados para interagir com recursos no servidor. *Get* é utilizado para obter um recurso específico do servidor, como um arquivo contendo informações sobre um livro ou uma lista de livros. *Post* é empregado para criar um novo recurso no servidor, como adicionar um novo livro a uma lista. *Put* serve para atualizar um recurso existente no servidor ou criar um novo caso ele não exista. *Delete* é usado para excluir um recurso específico do servidor (Mozilla Developer Network, 2024d).

JavaScript, uma linguagem de *script* amplamente utilizada, é projetada para criar interatividade em aplicações *web*, possibilitando a inclusão de elementos dinâmicos como animações, botões e menus. Essa linguagem permite funcionalidades avançadas, como colaboração em tempo real. No contexto de um ambiente hospedeiro, como uma aplicação *web*, oferece controle sobre os objetos do ambiente, enriquecendo a experiência do usuário. Além disso, JavaScript possui uma biblioteca padrão de objetos e um conjunto de elementos (Mozilla Developer Network, 2024c). Um objeto em

JavaScript é um tipo de dado composto por uma coleção de nomes ou chaves e valores, que podem conter qualquer tipo de dados, incluindo *strings*, números e *booleanos*, bem como métodos, que são funções contidas em um objeto (Rascia, 2018). Portanto, a flexibilidade de JavaScript é um dos seus pontos fortes.

No lado do cliente, JavaScript oferece objetos para controlar a aplicação *web* e o DOM, permitindo que aplicativos manipulem elementos HTML e respondam a eventos do usuário (Mozilla Developer Network, 2024a). Segundo o W3schools (2024a), o DOM é uma plataforma e interface, que facilita o acesso e a atualização dinâmica do conteúdo. Assim, o DOM, fundamental para tornar os sites interativos, é uma interface que possibilita a manipulação da estrutura e estilo de um site, permitindo que o JavaScript acesse e modifique os elementos do documento como objetos (Rascia, 2018).

JSX, por sua vez, é descrito como uma extensão de sintaxe para JavaScript, recomendada para ser utilizada em conjunto com o React, proporcionando a capacidade de construir a UI. Ele possui todas as funcionalidades do JavaScript, permitindo a criação de elementos do React, que são então renderizados para o DOM usando mecanismos como manipulação de eventos, alterações de estado ao longo do tempo e preparação de dados para exibição (React, 2024b). Dessa forma, o método *render* (), essencial no React em um componente de classe, é empregado para renderizar elementos no DOM (Rascia, 2018).

Por fim, o CSS é uma linguagem de estilo utilizada para definir a aparência de um documento escrito em HTML. Ela especifica como os elementos são exibidos na tela, focando na personalização e apresentação de aplicações *web*. (Mozilla Developer Network, 2024b)

4.3.1 REACTJS

O ReactJS é uma das bibliotecas de JavaScript mais populares para o desenvolvimento de aplicativos *web* e móveis, destacando-se pela sua eficiência e flexibilidade. Criado pelo Facebook, o React permite a construção de interfaces de usuário dinâmicas e reativas, facilitando a criação de componentes reutilizáveis. Sua abordagem baseada em um Virtual DOM otimiza a performance, tornando as atualizações mais rápidas e suaves. Além disso, é rico em ferramentas e bibliotecas complementares. Por essas razões, o ReactJS é amplamente adotado por desenvolvedores em todo o mundo. “O ReactJS é uma das bibliotecas de JavaScript

mais populares para desenvolvimento de aplicativos *web* ou para dispositivos móveis.” (L, Andrei 2023).

De acordo com Tania Rascia (2018), uma das características fundamentais do React é a capacidade de criar componentes personalizados e reutilizáveis, sejam eles componentes de classe ou funcionais. Isso permite a construção de interfaces de usuário de forma eficiente. O React facilita o armazenamento e a manipulação de dados por meio do uso de *state*, que representa dados que mudam com o tempo, e *props*, utilizados para transmitir dados de forma imutável entre componentes.

Ao construir uma interface com React deve-se dividir os componentes em pequenas partes, esses que serão gerenciados por um componente principal geralmente chamado de *App*. Um componente no React é uma parte independente, podendo ser algo simples como um botão ou até mesmo toda uma página (React, 2024a).

Conforme a documentação do React (React, 2024c), os componentes devem ser chaves únicas para inserir, excluir ou reordenar itens, garantindo flexibilidade e eficiência no desenvolvimento de interfaces.

Os componentes de classe implementam um método *render* () que recebe e retorna dados. Também é utilizado para atualizar um estado de um componente na interface, *useState* é uma função conhecida como *Hook*, que permite que componentes se gerenciem e tenham um estado (React, 2024e).

Como podemos ver na figura 2, esse código utiliza o React para criar um componente funcional chamado *Library*. Esse componente permite que o usuário adicione novos livros à biblioteca e utilize o *hook useState*, linha 1, para gerenciar o estado da aplicação.

Figura 2 – Exemplo de Código para adicionar livros

```

1  import React, { useState } from 'react';
2
3  const Library = ({ books, addNewBook, onUpdateBook, onDeleteBook }) => {
4      const [isModalOpen, setIsModalOpen] = useState(false);
5      const [bookToUpdate, setBookToUpdate] = useState(null);
6      const [newBook, setNewBook] = useState({
7          name: '',
8          author: '',
9          genre: '',
10         totalPages: 0,
11         currentPage: 0
12     });
13
14     const handleAddBook = () => {
15         const bookToAdd = { id: Date.now(), ...newBook };
16         addNewBook(bookToAdd);
17         closeModal();
18     };
19
20     const closeModal = () => {
21         setIsModalOpen(false);
22         setBookToUpdate(null);
23         setNewBook({ name: '', author: '', genre: '', totalPages: 0, currentPage: 0 });
24     };
25
26     return (
27         <div className="library">
28             <button onClick={() => setIsModalOpen(true)}>Add Book</button>
29             {isModalOpen && (
30                 <div className="modal">
31                     {/* Modal content */}
32                     <button onClick={handleAddBook}>Save Book</button>
33                 </div>
34             )}
35         </div>
36     );
37 };
38
39 export default Library;

```

Fonte: Elaborada pela autora (2024)

O React oferece recursos avançados, como estados internos nos componentes, que permitem que a interface do usuário reaja dinamicamente às interações do usuário ou a alterações nos dados. Utilizando uma renderização chamada Virtual DOM, uma representação em memória da estrutura da interface do usuário, que permite ao React atualizar apenas as partes modificadas da interface. Isso resulta em um desempenho mais rápido, proporcionando uma experiência de usuário mais responsiva e fluida. O desenvolvimento com React está intimamente relacionado à criação de uma experiência de usuário de alta qualidade, permitindo aos desenvolvedores criar interfaces dinâmicas

e intuitivas, isso possibilita aos desenvolvedores criar interfaces altamente personalizáveis e adaptáveis, ajustando-se às preferências individuais dos usuários (React, 2024a).

O React oferece vantagens que JavaScript não proporciona da mesma maneira simples, especialmente para a construção de interfaces gráficas dinâmicas e interativas. Enquanto JavaScript é a base para o desenvolvimento *web* (W3schools, 2024b), React é uma biblioteca que torna a construção de interfaces de usuário mais eficiente e organizada que se pode criar componentes reutilizáveis, que facilitam a manutenção e escalabilidade do código (React, 2024f). Em JavaScript, essas funcionalidades exigiriam muito mais código e complexidade para alcançar o mesmo resultado.

Em suma, o React é uma ferramenta versátil para o desenvolvimento de aplicações, oferecendo aos desenvolvedores as ferramentas necessárias para criar interfaces de usuário modernas, eficientes e altamente personalizáveis.

4.3.1.1 Experiência do Usuário (UX)

A *User Experience* se trata de todo o processo de interação e percepção que um usuário tem ao utilizar uma aplicação. Não se trata apenas de funcionalidade ou usabilidade, mas envolve aspectos emocionais e psicológicos que influenciam como os usuários se sentem durante a interação com a aplicação ou aplicativo. Deve compreender as necessidades, expectativas e comportamentos dos usuários, garantindo que suas experiências sejam positivas e satisfatórias, como a facilidade de navegação e a eficiência (Santos, 2021).

Para criar uma experiência de usuário de alta qualidade, o UX emprega uma variedade de técnicas, fundamentos e metodologias, incluindo pesquisa de usuários, testes de usabilidade, prototipação e design iterativo. O objetivo é desenvolver interfaces intuitivas, que atendam às expectativas dos usuários e supram suas necessidades de forma eficaz, proporcionando uma experiência satisfatória. Quando bem executada, essa abordagem resulta em maior engajamento e satisfação do cliente, contribuindo significativamente para o sucesso da aplicação.

Ao utilizar o React no desenvolvimento de aplicações, os desenvolvedores podem empregar técnicas e metodologias de UX, como pesquisa de usuários, testes de usabilidade e *design* iterativo. Isso permite que eles compreendam melhor as

necessidades, expectativas e comportamentos dos usuários, garantindo que a aplicação forneça uma experiência satisfatória e eficaz.

4.3.1.2 Interface do Usuário (UI)

A *User Interface* desempenha um papel essencial no desenvolvimento *front-end*, visto que é responsável por criar a experiência visual e interativa que os usuários têm ao interagir com uma aplicação. A UI cuida de elementos visuais e interativos, incluindo botões, formulários, menus, ícones, gráficos e muito mais. Seu objetivo principal é proporcionar uma experiência de usuário intuitiva e eficiente, garantindo que a interface seja facilmente compreensível e utilizável (Ribeiro, 2023).

No contexto do desenvolvimento *front-end*, o React desempenha um papel crucial na implementação da UI. Afinal, uma das principais vantagens do React é sua abordagem baseada em componentes. Ao utilizar o React para criar componentes de UI, os desenvolvedores podem garantir uma interface consistente e coesa em toda a aplicação. "A meta primária do React é minimizar os erros que ocorrem quando os desenvolvedores estão construindo UIs. Isto é devido ao uso de componentes - autocontidos, partes lógicas de códigos que descrevem uma parte da interface do usuário" (Mozilla Developer Network, 2024a).

Os componentes podem ser facilmente personalizáveis e combinados para criar um *layout* complexo, proporcionando uma experiência de usuário fluida e intuitiva. Além disso, o React oferece recursos avançados, como o uso de estados internos nos componentes, permitindo que a interface do usuário reaja dinamicamente às interações do usuário ou a alterações nos dados (React, 2024e).

Além de garantir a usabilidade, a UI também desempenha um papel fundamental na estética do *layout*. Ao adotar um *design* visual atrativo e alinhado com o tema da aplicação, os desenvolvedores podem criar uma experiência visualmente harmoniosa e agradável para os usuários. Isso inclui a cuidadosa seleção de cores, tipografia, ícones e espaçamento, visando criar uma interface atraente e fácil de usar.

Logo, o React é uma ferramenta importante para o desenvolvimento de interfaces de usuário, permitindo aos desenvolvedores criar experiências visuais e interativas de alta qualidade. Ao utilizar componentes React e práticas de *design* responsivo, os desenvolvedores podem garantir uma UI eficiente e esteticamente

agradável para os usuários, contribuindo significativamente para o sucesso de um projeto de software

5 METODOLOGIA

Nesta seção, serão apresentadas as etapas e procedimentos adotados para alcançar os objetivos propostos neste estudo para o desenvolvimento da interface da aplicação de organização pessoal de leitura.

5.1 Modelo de Desenvolvimento

A metodologia adotada para o desenvolvimento deste projeto: a implementação de uma aplicação organizadora de leitura pessoal, nomeada como BookMark, baseou-se na abordagem do modelo de prototipação. Portanto, as etapas de desenvolvimento são as seguintes: comunicação, plano rápido, modelagem, construção do protótipo e implementação.

5.2 Comunicação

O primeiro passo para criar um protótipo é identificar as necessidades que serão resolvidas nesse processo. Isso envolve pensar nas demandas do usuário e em quais problemas eles enfrentam, além de como a solução poderá resolvê-los.

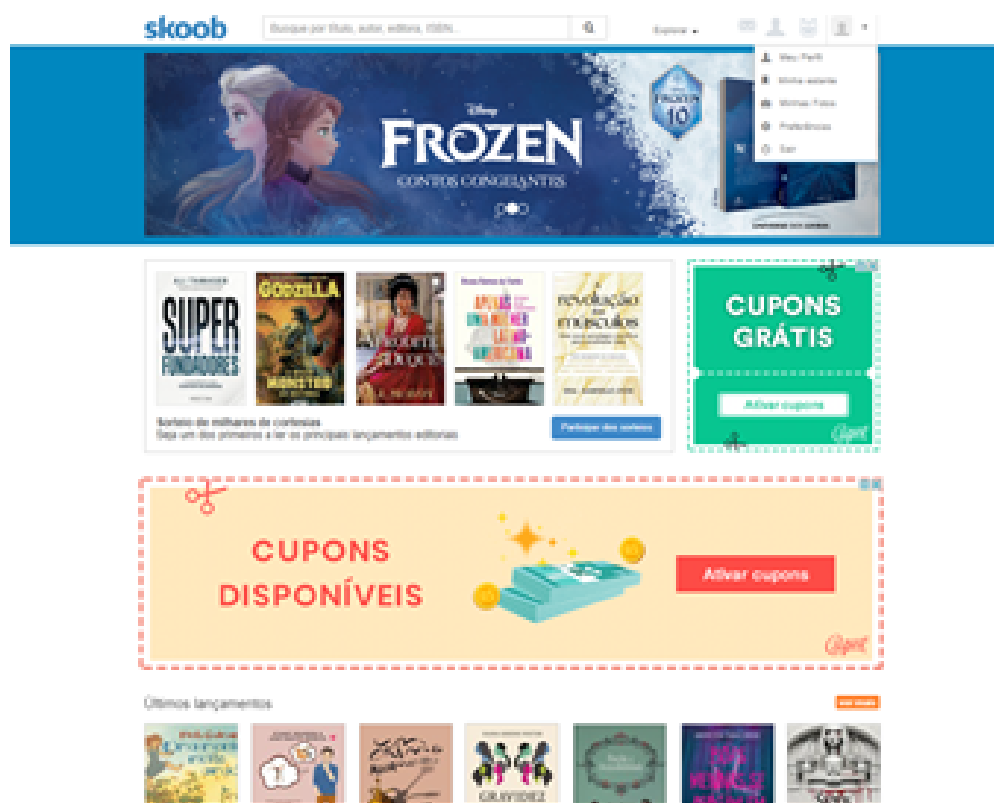
O levantamento de requisitos é uma etapa crucial do processo de desenvolvimento, pois define as bases sobre as quais o projeto será construído.

5.2.1 SKOOB

O Skoob se identifica como a maior rede social para leitores do Brasil. Funciona como uma estante virtual, onde você pode não só colocar os livros que já leu, como também aqueles que ainda deseja ler. Tudo de forma organizada para que você não se perca durante as leituras. E você ainda tem a vantagem de poder compartilhar suas opiniões com seus amigos, fazer trocas de livros, participar de sorteios, ganhar cortesias e muito mais (Skoob, 2024).

A análise da página inicial do Skoob, Figura 3, revela uma série de aspectos que impactam a experiência do usuário de forma significativa. Ao acessar a aplicação, em sua página inicial os usuários são recebidos por uma quantidade massiva de anúncios, resultando em uma poluição visual que pode distrair e sobrecarregar os visitantes.

Figura 3 - Página Inicial Skoob.



Fonte: Skoob (2024)

Há uma falta de hierarquia visual que torna difícil para os usuários identificarem os elementos mais importantes da página. O conteúdo na totalidade está muito centralizado na página, deixando as laterais com um espaço vazio. Existem diversas listas distribuídas pela página com o mesmo tamanho, como a lista de sorteios, lançamentos e livros mais procurados. O principal ponto é a ausência da lista de leitura do próprio usuário, que é uma lacuna notável, pois não oferece uma visão personalizada e relevante do conteúdo em si.

A organização do menu superior em dois menus expansíveis pode dificultar a navegação eficiente, especialmente para novos usuários. Diversos estudos iniciais sobre exames visuais de páginas *web* sugerem que os usuários decidem muito rapidamente em quais partes da página provavelmente encontrarão informações úteis, e então quase não olham mais as outras partes - quase como se elas não estivessem lá (Krug, 2014).

Ressaltada a importância de garantir que as informações cruciais sejam apresentadas de forma proeminente e facilmente acessível aos usuários, evitando que se percam em áreas menos visíveis da interface. No entanto, na aplicação Skoob, a presença de uma barra inferior com funcionalidades importantes que acaba por ficar

escondida no final da página, longe do olhar do usuário, prejudica tanto a usabilidade quanto a experiência, já que o usuário não encontrará aquelas informações importantes a menos que as procure ativamente.

Não só isso, mas a falta de um botão claro para adicionar livros à biblioteca do usuário pode causar confusão e frustração. Afinal, para um usuário adicionar um livro, ele deve buscar na barra de pesquisa e entrar na página daquele livro específico, um fluxo longo e anti-intuitivo, o que não é bom para uma aplicação, pois segundo Krug (2014), "Como regra geral, as pessoas não gostam de ter que descobrir como fazer algo. O fato das pessoas que criaram a aplicação não terem se importado com deixar as coisas óbvias - e fáceis - pode diminuir nossa confiança na aplicação e seus donos."

Quanto ao *front-end*, o uso predominante do *layout* branco pode causar cansaço visual. A página do livro no Skoob, figuras 4 e 5, apresentam uma densidade de informações significativa, o que pode sobrecarregar visualmente alguns usuários. Uma observação importante é a duplicidade do nome do livro, o que pode parecer redundante e ocupar espaço desnecessário na tela.

Um aspecto importante a ser considerado é a distinção entre textos e *links* na figura 4, que pode não ser suficientemente clara, tornando a identificação dos elementos interativos mais difícil para os usuários. Embora o menu lateral seja disposto à esquerda, o que é uma prática comum e auxilia na navegação da aplicação, algumas seções do menu possuem funcionalidades redundantes. Por exemplo, a presença de opções como "Sinopse" e "Edições" no menu lateral pode ser desnecessária, considerando que essas informações já estão disponíveis na página central. O mesmo se aplica à seção de vídeos. Esses aspectos indicam um possível desperdício de recursos e contribuem para uma sensação de poluição visual na tela, o que pode afetar negativamente a experiência do usuário.

Figura 4 - Página de Livro Skoob

skoob Busque por título, autor, editora, ISBN... Explorar

Gente ansiosa

4.3 19.414 avaliações

LERAM 21.680 LENDO 2.744 QUEREM LER 34.948 RELENDO 28 ABANDONOS 1.299 RESENHAS 5.114

Favorites (4.678) Desejados (2.617) Trocam (7) Avaliaram (19.414)

Best-seller instantâneo do New York Times, o novo romance do autor de Um homem chamado Ove é um "romance engraçado e emocionante".

A busca por um apartamento não costuma ser uma situação de vida ou morte, mas uma visita imobiliária toma tais dimensões quando um fracassado assaltante de banco invade o apartamento e faz de reféns um grupo de

[ver mais](#)

Gente ansiosa
Fredrik Backman

Nenhuma oferta encontrada

ISBN-13: 9786555321111
ISBN-10: 6555321113
Ano: 2021 / Páginas: 368
Idioma: português
Editora: Rocco

★★★★★ 5

Lido +

Compartilhe

Sinopse

Edições 3

Vídeos 12

Grupos 0

Resenhas 5.114

Edições (3) [ver mais](#)

Similares (15) [ver mais](#)

Resenhas para Gente ansiosa (5.114) [ver mais](#)

Coloque-se no lugar do outro, e decida não roubar um banco.
Heliton ★★★★★
em 17/10/21

É um fato que a empatia é uma capacidade que faz parte da natureza do ser humano. Mas muitos podem dizer que só faz parte mesmo, pois há quem esqueça de usá-la na maioria das vezes. Nunca vou esquecer desse grupo de reféns, isso é um outro fato também. Sabe, há momentos durante a sua vida em que alguns sentimentos surgem ou coisas acontecem onde você acha melhor guardar para si a falar para alguém, talvez por medo de ser julgado, de não estar confortável ou vergonha em dizer, e outr... [leia mais](#)

Fonte: Skoob (2024)

Figura 5 - Página de Livro Skoob

Gente ansiosa
Fredrik Backman - 2021 / 368 páginas - Rocco

Tags de Estante e Pública

Estante: (use tags para ajudar a categorizar o livro na sua estante.)
Ex.: ganhei, pesquisa, estudo, vou comprar, presente.

Públicas: tags que ajudem na busca do livro por outros usuários.
Ex.: romance, vampiros, grécia, triste, best-seller.

Data de leitura
Informe a data em que leu o livro, caso não se lembre mais, coloque uma data aproximada. Essa data irá definir a ordem em que aparecerão seus livros na lista de lidos. Os mais recentes primeiro.

Dia: Mês: Ano:

[Salvar alterações](#)

Minha cópia

Resenha

Histórico de Leitura

Capa / Edição

Editar para Troca

LV HQ RV
PA EB AB

Fonte: Skoob (2024)

No entanto, a presença de elementos coerentes contribui para a coesão visual da página, proporcionando uma experiência mais consistente para o usuário. A capa do livro é um dos elementos mais proeminentes na tela, destacando-se como o segundo maior elemento e incorporando uma faixa azul que atrai a atenção para detalhes relevantes da leitura de outros usuários. Isso cria uma dinâmica visual interessante e pode incentivar os usuários a explorarem mais sobre o livro.

A presença de elementos típicos de redes sociais, como resenhas de usuários, vídeos e estatísticas, sugere uma abordagem com foco na comunidade, incentivando a interação e o engajamento dos usuários, porém não há alternativas para pessoas que não tem interesse no aspecto social e apenas na organização de leitura.

A nota em estrelas e o status do livro são intuitivos, facilitando a compreensão do usuário sobre a avaliação e a situação atual da leitura. Por outro lado, a falta de clareza nos detalhes do livro pode representar um desafio para os usuários, uma vez que o ícone de adição (+) pode não ser imediatamente identificado como um botão para acessar essas informações.

Da mesma forma, os botões de favorito, propriedade, desejo, empréstimo, troca e metas, apesar de apresentarem ícones, dependem da interação do usuário para revelar sua funcionalidade por meio de texto ao passar o mouse sobre eles. Isso pode exigir um esforço adicional por parte do usuário para entender completamente as opções disponíveis, o que pode afetar a usabilidade e a experiência geral do usuário.

A página de lista de leitura do Skoob, figura 6, apresenta uma navegação por abas que direciona o foco do usuário para a estante, oferecendo uma organização eficiente dos conteúdos. A barra lateral se destaca pela presença de ícones e textos, o que contribui para uma experiência intuitiva de navegação, permitindo que os usuários filtrem e personalizem suas visualizações de acordo com suas preferências. A quantidade de itens em cada categoria é um adicional interessante por oferecer uma visão geral das informações disponíveis.

Figura 6 - Página de Livros Skoob

Fonte: Skoob (2024)

Os livros são apresentados em forma de galeria, dispostos pela capa do livro, o que facilita a identificação visual e rápida dos títulos. Informações importantes, como status de leitura, nome do livro, autor e classificação por estrelas, são exibidas de maneira clara e acessível, oferecendo aos usuários uma visão abrangente de sua lista de leitura.

Esta página se destaca como uma das melhores da aplicação, pois cumpre seu propósito central com foco nos livros e em funcionalidades úteis para os usuários. Ao mesmo tempo, não deixa de oferecer recursos relacionados à rede social, que podem ser acessados pelos interessados por meio das abas disponíveis. Essa abordagem equilibrada entre funcionalidade e interatividade contribui para uma experiência de usuário satisfatória e engajadora na aplicação Skoob.

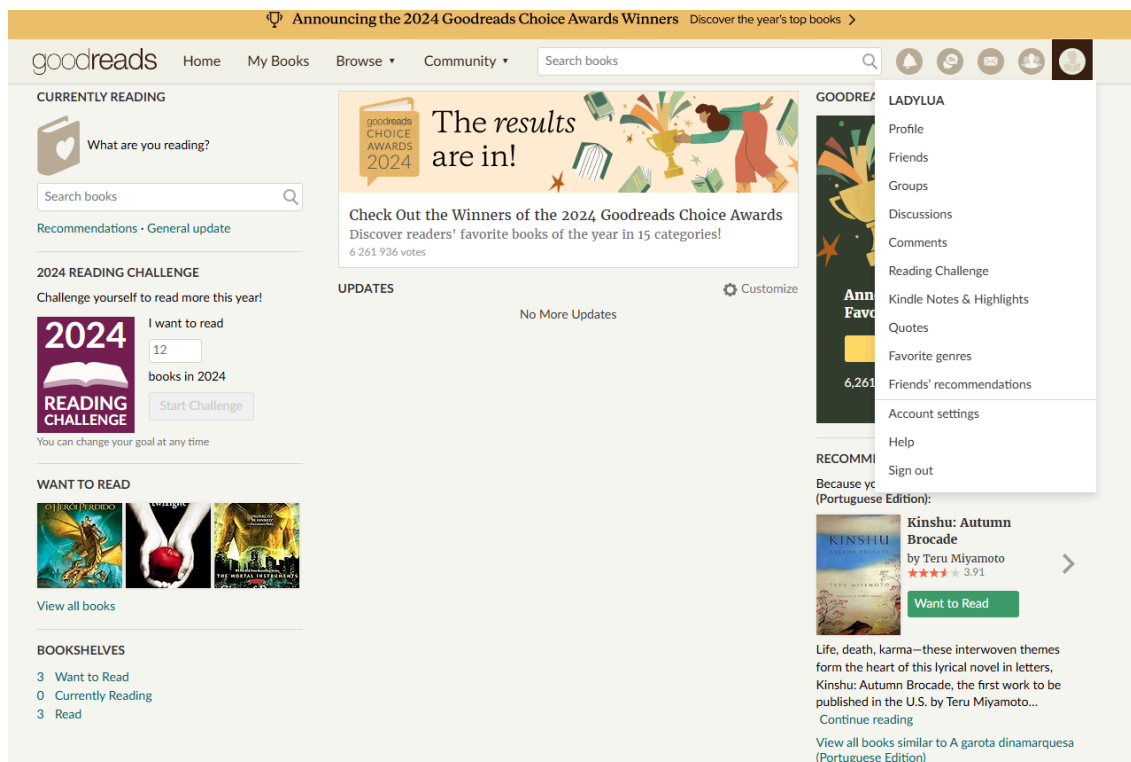
5.2.2 Goodreads

O Goodreads é uma aplicação amplamente utilizada por leitores ao redor do mundo, oferecendo diversas funcionalidades que enriquecem a experiência de leitura. Fundado em 2007, o Goodreads se destaca como uma plataforma social onde os usuários podem avaliar livros, escrever resenhas, e criar bibliotecas virtuais para organizar suas leituras passadas, presentes e futuras (Goodreads, 2024).

A página inicial do Goodreads, apresentada na figura 7, possui muitos textos sem hierarquia clara, a aplicação dificulta a compreensão e a navegação para os

usuários. "Quanto tudo na página está chamando minha atenção, o efeito pode ser esmagador." (Krug, 2014) Esse trecho pontua que uma falta de ponto focal na tela contribui para uma sensação de sobrecarga visual, onde o usuário recebe tantas informações que não sabe por onde começar a navegar.

Figura 7 - Página Inicial Goodreads



Fonte: Goodreads (2024)

A falta de coesão nos elementos correlatos, como as capas de livro exibidas em diferentes formatos, pode causar confusão e prejudicar a consistência visual da página. A presença de diferentes formatos para as capas de livro pode levar os usuários a interpretarem erroneamente a informação apresentada, já que não há uma uniformidade visual que indique claramente a relação entre esses elementos. Além disso, as partes laterais da página diferem da parte central, mas essa discrepância pode passar despercebida devido aos tamanhos semelhantes dos elementos, o que pode confundir ainda mais os usuários e dificultar a compreensão da organização da página. Essa falta de coesão visual compromete a experiência do usuário, tornando-a menos intuitiva e coesa.

Assim como no Skoob, a falta de um botão claro para adicionar livros à biblioteca do usuário é uma falha significativa.

O fato de a aplicação estar em inglês pode representar obstáculos à acessibilidade para os usuários brasileiros, limitando seu uso e compreensão. E a presença de fontes pequenas torna a aplicação menos acessível para pessoas com baixa visão. Por outro lado, o Goodreads parece atender melhor ao seu papel como rede social, apresentando uma *time-line* central na página inicial, o que facilita a interação e a descoberta de novos conteúdos pelos usuários, todavia a quantidade de posts acaba desnivelando a interface e atrapalhando o layout.

Além disso, a escolha de uma cor mais neutra e agradável em comparação com o Skoob pode contribuir para uma experiência visual menos cansativa para os usuários.

Na página do livro no Goodreads, figura 8, a capa do livro assume uma posição de destaque evidente, capturando imediatamente a atenção do usuário com uma apresentação visualmente atraente, composta por uma sombra esfumada que adiciona profundidade e foco à imagem. Este design permite que os usuários identifiquem facilmente o livro em questão e estabeleçam uma atmosfera atraente para explorar mais sobre a obra. Além disso, o nome do livro é apresentado de forma proeminente logo ao lado da capa, garantindo uma identificação clara e imediata do título.

Entretanto, existe uma falta de clareza na identificação de elementos interativos, por exemplo, o texto "Avalie esse livro" posicionado abaixo das estrelas de classificação pode não ser imediatamente reconhecido como um link para acessar uma avaliação mais detalhada do livro. Apesar disso, na figura 8, o status de leitura é apresentado de forma clara e direta, permitindo que os usuários visualizem rapidamente o seu progresso na leitura do livro, porém diferente do Skoob não altera a cor do status, o que pode dificultar uma identificação, que contribuiria para uma experiência de usuário mais intuitiva e eficiente, uma vez que o acompanhamento do progresso de leitura é uma funcionalidade essencial para os usuários.

Figura 8 - Página de Livro Goodreads



The screenshot shows the Goodreads interface for the book 'Gente ansiosa' by Fredrik Backman, translated by Carmem Montes Cano. The page features a book cover on the left, a title and author section, a star rating of 4.18, a synopsis, and various genre tags. The layout is clean and organized, with a clear hierarchy of information.

Goodreads The Big Books of Spring Discover readers' most anticipated new releases this season >

Lar Meus livros Navegar Comunidade Pesquisar livros

FREDRIK BACKMAN
GENTE ANSIOSA
uma novela

Gente ansiosa
Fredrik Backman, Carmem Montes Cano (Tradução)

★★★★☆ 4.18 627.636 avaliações · 76.822 avaliações

Gente é ansiosamente uma comédia desenfrenada sobre um robô caído que termina em um grupo de pessoas sendo mantidas cautelosamente durante a jornada de portas abertas de um apartamento.

Uma só má ideia: uma má ideia de verdade. Não há falta mais.

Mostre mais ▾

Gêneros **Ficção** Contemporâneo Áudio-livro Mistério Humor Adulto Ficção Literária ...mais

400 páginas, brochura

Publicado pela primeira vez em 25 de abril de 2019

Prêmios literários [Indicado ao Goodreads Choice Award de Ficção \(2020\)](#) [Indicado ao Prêmio Livro do Mês do Ano \(2020\)](#) [Indicado à Lista de Leitura RUSA CODES para Ficção de Relacionamento \(2021\)](#) [Ficção Criminal em Tradução Indicado à Adaga por Neil Smith \(2021\)](#)

Título original [Folk com melhor](#)

Contexto [Suécia](#)

Quero ler ▾

Compre na Amazon BR ▾

☆☆☆☆
Avalie este livro

Fonte: Goodreads (2024)

Em relação à organização das informações sobre o livro, na figura 9 a disposição vertical dos elementos e a aplicação de uma hierarquia visual adequada garantem que a tela não pareça sobrecarregada ou desorganizada. Mesmo com uma quantidade considerável de informações disponíveis, a apresentação ordenada e claramente estruturada facilita a compreensão e a navegação dos usuários pela página. Por fim, a seção de rede social, que inclui comentários e resenhas, é apresentada de forma mais discreta em comparação com o Skoob por se localizar mais ao final da página. Essa abordagem mais sutil pode ser apreciada por usuários que preferem uma experiência menos focada em interações sociais e mais centrada na descoberta de novas obras e na gestão pessoal da leitura.

Figura 9 - Página de Livro Goodreads

Gente Ansiosa (Paperback)
por Fredrik Backman (autor do Goodreads)
Alterar edição

Minha classificação: ★★★★★ Claro

Estantes/etiquetas: Escolha prateleiras... legal, leia

O que você acha? Dicas de formatação Inserir livro/autor | Ampliar campo de texto: |

Digite sua avaliação (opcional)

☐ Ocultar a crítica inteira por causa de spoilers

Datas lidas
Retendo?
Agora você pode acompanhar todas as vezes que leu um livro. Certifique-se de preencher o ano final para adicioná-lo ao seu Desafio de Leitura!

Data de início (opcional) Ano ▼ Mês ▼ Dia ▼ Definir para hoje Data de término (opcional) Ano ▼ Mês ▼ Dia ▼ Definir para hoje

Adicionar data de leitura

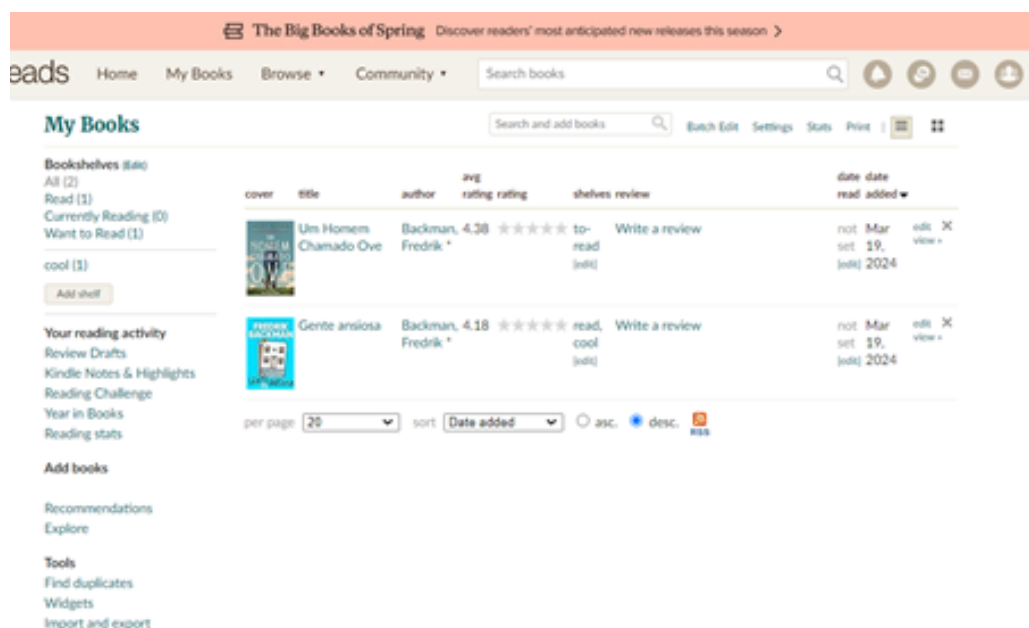
Mais detalhes v

Publicar ☐ Postar no blog ☒ Adicionar ao meu feed de atualização

Fonte: Goodreads (2024)

A página de lista de leitura do Goodreads, figura 10, apresenta uma barra lateral que, embora forneça acesso a diferentes recursos da aplicação, pode parecer um pouco sobrecarregada, especialmente com elementos que não estão diretamente relacionados à lista de leitura, como a seção "Ferramentas". A lista de livros apresenta detalhes essenciais, como capa, título, autor, nota média, classificação por estrelas e datas de inclusão ou leitura. O Goodreads também permite visualizar a lista no formato de galeria, oferecendo uma experiência mais visual para quem prefere esse estilo. Além disso, os usuários podem personalizar a exibição, escolhendo quais informações desejam ver. E criar listas além das pré-criadas. No entanto, o layout da página é frequentemente criticado por sua aparência antiquada e poluída, o que pode afetar negativamente a experiência do usuário. A falta de um design mais moderno pode reduzir o apelo visual da página e dificultar a navegação e interação dos usuários com a lista de leitura.

Figura 10 - Página de Livros Goodreads

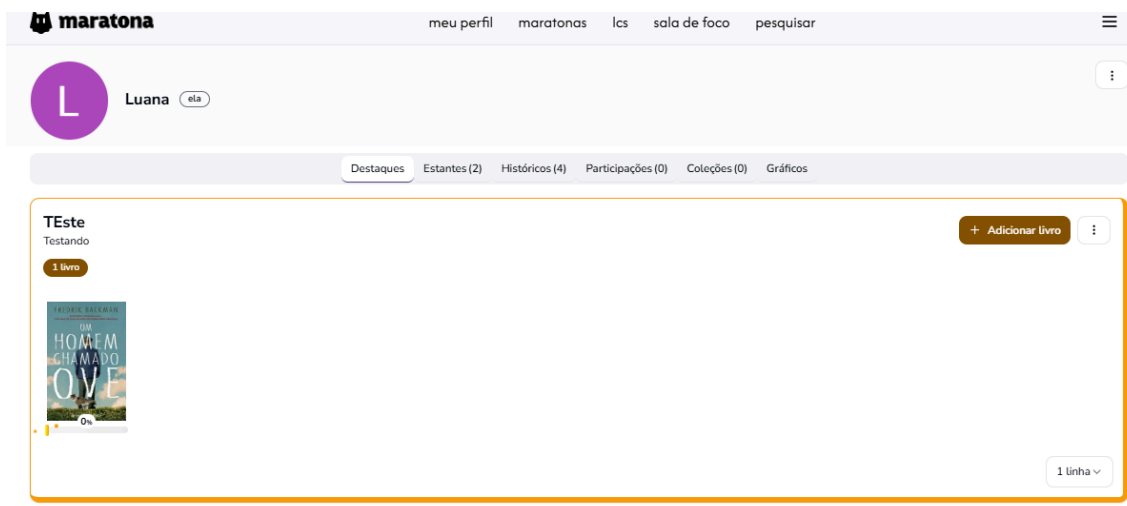


Fonte: Goodreads (2024)

5.2.3 Maratona

A Maratona oferece aos usuários uma visão abrangente de suas atividades de leitura, permitindo o acompanhamento do progresso individual. Por meio do Maratona, uma aplicação de organização de leitura, os leitores podem se conectar ainda mais, aproveitando funcionalidades como maratonas de leitura e leituras coletivas. O Maratona possui o layout mais simples e menos carregado que o Skoob e Goodreads como é possível observar na figura 11.

Figura 11 - Página Inicial Maratona



Fonte: Maratona (2024)

A página inicial do Maratona se destaca por sua simplicidade e foco na lista de livros adicionados recentemente, além disso, a presença de um botão claramente identificável para adicionar livros à lista de leitura, proporcionando uma experiência mais direta e personalizável para os usuários. Ao priorizar os livros recentemente adicionados e oferecer a opção de filtrar por livros em andamento, o Maratona busca atender às necessidades individuais dos usuários e fornece um ambiente mais adaptável às suas preferências de leitura.

Na página do livro no Maratona, figura 12, observa-se uma abordagem semelhante ao Goodreads, com a capa do livro ocupando uma posição de destaque, enfatizando sua importância como ponto de partida para a exploração da obra. A presença de um efeito de sombreamento suave por trás da capa ajuda a direcionar o foco do usuário para o elemento central da página, criando uma apresentação visualmente atraente e coesa.

Uma característica a se notar é a ausência de um menu lateral tradicional, substituído por um botão de "adicionar livro", que, embora simplifique a navegação inicial, pode gerar confusão devido à falta de feedback visual após a adição do livro ou de um livro já adicionado. A falta de alteração no botão após a inclusão do livro pode induzir os usuários a adicionarem o mesmo livro várias vezes, o que pode causar confusão e duplicação de dados.

A hierarquia de informações é estabelecida de maneira eficaz, com o título do livro ocupando uma posição proeminente, seguido pelo autor e sinopse. Os gêneros são apresentados em tags com cores de fundo distintas, destacando-se na página e contribuindo para a coesão visual dos elementos.

A seção inferior da página exibe uma variedade de informações adicionais, como edições do livro, leituras coletivas e estantes públicas, todas organizadas em tags retangulares com cores diferentes para distinguir cada categoria.

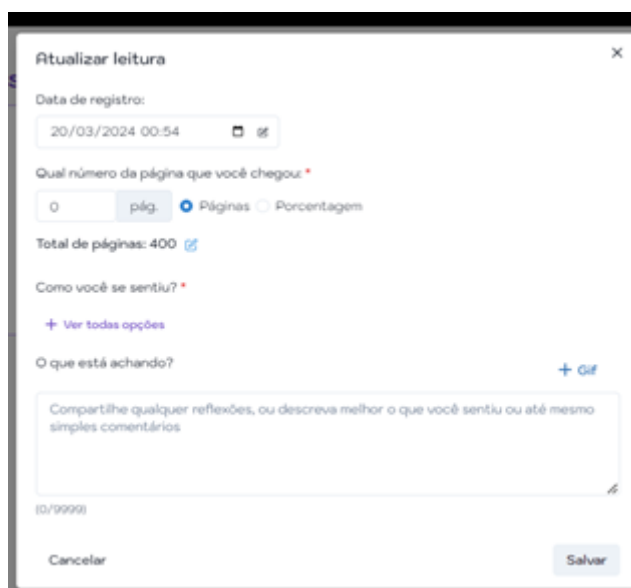
Figura 12 - Página de Livro Maratona



Fonte: Maratona (2024)

No entanto, na figura 13 pode-se observar que a falta de opções para adicionar informações de leitura diretamente na página do livro pode representar uma limitação para os usuários, exigindo que eles naveguem para outras áreas da aplicação para realizar essa ação. Além disso, não há uma opção de atualização manual de *status*, já que as atualizações ocorrem apenas quando a página lida é alterada automaticamente. Isso pode resultar em uma experiência menos dinâmica para os usuários, especialmente aqueles que desejam registrar seu progresso de leitura imediatamente.

Figura 13 - Página de Livro Maratona



Fonte: Maratona (2024)

Outro aspecto a ser considerado é o método de avaliação de livros na figura 14, que difere do formato tradicional de avaliação com estrelas. Em vez disso, os usuários são apresentados a um conjunto de tags pré-definidas para expressar suas emoções em relação à obra. Embora essa abordagem possa ser considerada inovadora, pode desagradar os usuários acostumados ao método convencional de avaliação por estrelas.

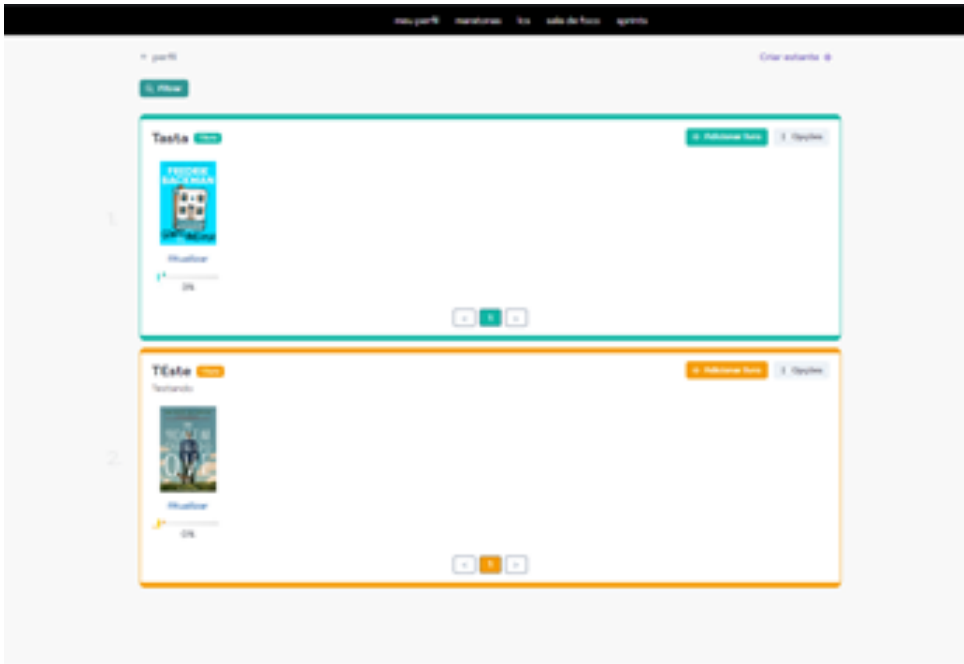
Figura 14 - Página de Livro Maratona



Fonte: Maratona (2024)

Na página de lista de leitura do Maratona, figura 15, destaca-se a possibilidade de personalização, permitindo aos usuários criar e gerenciar diversas listas de leitura de forma flexível. A capacidade de alterar a cor das listas contribui para uma organização visual eficaz, facilitando a distinção entre diferentes seções e tornando a página mais agradável esteticamente. Além disso, a exibição da quantidade de livros em cada lista oferece uma visão rápida e clara do conteúdo de cada seção, auxiliando na navegação e no gerenciamento do acervo pessoal.

Figura 15 - Página de Livros Maratona



Fonte: Maratona (2024)

Ao contrário de outras aplicações, como Skoob e Goodreads, que podem apresentar menus laterais sobrecarregados, a página de lista de leitura do Maratona adota uma abordagem mais simplificada. Em vez disso, é fornecido um botão de filtro intuitivo, que permite aos usuários realizarem buscas rápidas por nome, autor, favoritos, recomendações e outras categorias relevantes. As informações essenciais do livro, como progresso de leitura, capa, autor e nome, são apresentadas de forma clara e concisa, priorizando a experiência centrada na lista de leitura do usuário.

A seguir a tabela 1 apresenta uma comparação entre as aplicações analisadas.

Quadro 1 - Análise de sites

Sites	Skoob	Goodreads	Maratona
Poluição visual	Grande	Média	Pequena
Botão Adicionar Livro	Não	Não	Sim
Cadastrar Livros	Sim, mas qualquer um pode cadastrar o mesmo livro várias vezes.	Sim, mas deve fazer parte de um grupo específico	Sim, mas deve pedir o cadastro via email

Alterar informação do livro	Sim	Não	Não
Dados do Livro	Capa, Nome, Autor, ISBN, Sinopse, Ano, Páginas, Editora, Nota média, Idioma	Capa, Nome, autor, tradutor, Nota média, Sinopse, Gêneros, Páginas, Tipo de Livro, Ano, Idioma	Capa, Nome, Autor, Páginas, Editora, Ano, Idioma, Sinopse, Gráfico de emoções
Dados de Leitura do usuário	Estrelas, <i>Status</i> , Favorito, Desejo, Propriedade, Empréstimo, Troca, Meta, Resenha, Estante, Tags, Data de Leitura e Tipo	Estrelas, Estante, Resenha, Data Início, Data término, Releitura.	Estante, Data de leitura, Páginas Lidas, Sentimentos, Resenha, Histórico de leitura, Avaliação por <i>tag</i>
Detalhes do Livro	Capa, Nome, Autor, Editora, Estrelas, <i>Status</i>	Capa, Nome, Autor, Estrelas média, Estrelas, Lista, Resenha, Data lida, Data Adicionada	Capa e Progresso
Personalizável	Não	Sim, pode-se criar novas listas	Sim, pode-se criar novas listas e personalizar a cor de cada uma

Fonte: Elaborado pela autora (2024)

Nesse sentido, a análise de aplicações correlatas serviu como ponto de partida para compreender as necessidades e expectativas dos usuários, bem como identificar lacunas ou oportunidades de melhoria em relação às aplicações existentes, como Skoob, Goodreads e Maratona. Além disso, os objetivos definidos para o projeto orientam a seleção e priorização dos requisitos levantados, garantindo que a solução desenvolvida atenda às expectativas dos usuários e aos objetivos do próprio projeto. É importante considerar diferentes perspectivas e contextos de uso ao definir esses requisitos, levando em conta não apenas as funcionalidades desejadas, mas também os aspectos relacionados à usabilidade, acessibilidade e experiência do usuário.

A seguir a tabela 2 apresenta os requisitos funcionais levantados.

Quadro 2 - Requisitos funcionais

Código	Descrição do Requisito
RF01	Permitir cadastrar livros
RF02	Permitir atualizar livros

RF03	Permitir criação de Lista de leitura
RF04	Permitir adição e remoção de livros da lista de leitura
RF05	Gerar progresso do livro
RF06	Gerar gênero de leitura
RF07	Disponibilizar calendário de leitura com destaque para intervalo entre início e término de leitura
RF08	Permitir a criação de Meta de leituras
RF09	Gerar progresso de Metas
RF10	Permitir personalização do <i>layout</i>
RF11	Confirmação de senha do usuário
RF12	Método de recuperar senha

Fonte: Elaborado pela autora (2024)

A tabela 3 apresenta os requisitos não funcionais levantados.

Quadro 3 - Requisitos não funcionais

Código	Descrição do Requisito
RNF01	Responsividade
RNF02	Intuitividade no <i>layout</i>
RNF03	O sistema deve ser acessível, atendendo a padrões de acessibilidade

Fonte: Elaborado pela autora (2024)

No atual projeto, os requisitos levantados abrangem diversas funcionalidades, como o layout, cadastro de livros, listas de leitura, estatísticas e calendário. O layout visualmente confortável e personalizável, por exemplo, visa oferecer aos usuários a possibilidade de adaptar a interface de acordo com suas preferências estéticas, proporcionando uma experiência mais agradável. O cadastro de livros, por sua vez, inclui informações relacionadas aos livros, como nome, autor, gêneros, páginas totais, páginas lidas, progresso de leitura, entre outros.

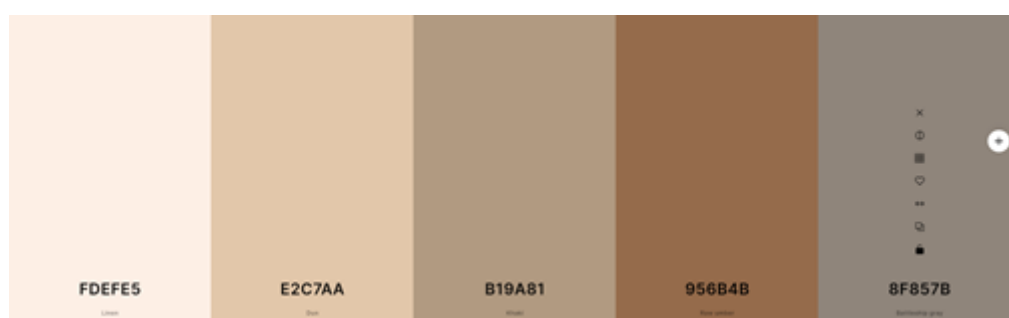
Esses dados são essenciais para que os usuários possam gerenciar sua biblioteca pessoal de forma eficiente e organizada, facilitando o acompanhamento do progresso de leitura e o acesso a informações relevantes sobre cada livro. As listas de leitura e as estatísticas de leitura fornecem aos usuários ferramentas adicionais para organizar e analisar seus hábitos de leitura, permitindo a criação de listas personalizadas e o

acompanhamento de métricas como número de livros lidos, gêneros preferidos. Já o calendário que colore os dias no intervalo entre o início e o término contribui para a visualização e planejamento do progresso de leitura ao longo do tempo, ajudando os usuários a manterem-se motivados e comprometidos com suas metas de leitura.

5.3 Plano Rápido

O design e o projeto de interface são essenciais para o desenvolvimento de uma aplicação, especialmente quando se trata de um projeto voltado para organização pessoal de leitura. A escolha de cores, fontes, imagens e elementos visuais é fundamental para criar uma atmosfera agradável e funcional. A escolha da paleta de cores desempenha um papel crucial na interface de uma aplicação. A paleta de cores proposta para a interface da aplicação de organização pessoal de leitura apresenta uma abordagem cuidadosamente selecionada que pode ser analisada na figura 16.

Figura 16 - Paleta de cores



Fonte: Coolors.co (2024)

A cor creme #FDEFE5, escolhida como a base da aplicação, é caracterizada por sua suavidade e neutralidade, transmitindo uma sensação acolhedora e convidativa aos usuários. Essa tonalidade suave cria um fundo que não apenas destaca o conteúdo da aplicação, mas também reduz a fadiga visual, promovendo uma experiência de navegação confortável e relaxante.

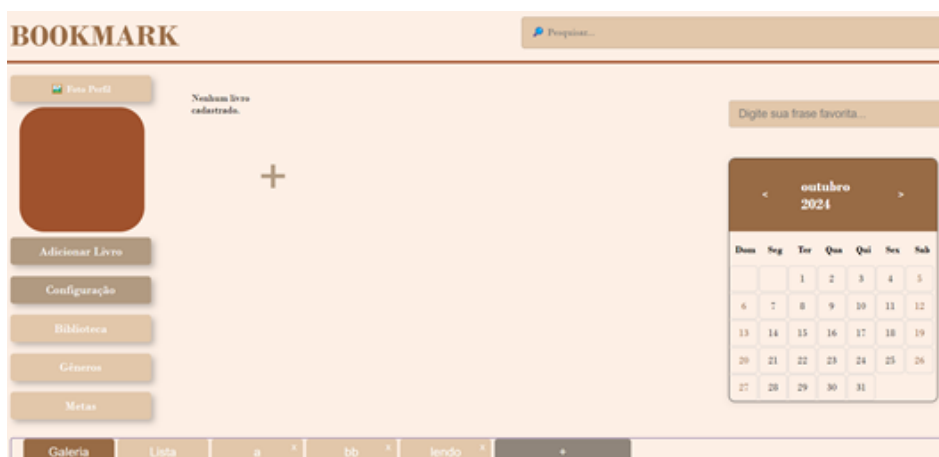
As cores complementares, como marrom-claro #E2C7AA, marrom #B19A81 e marrom escuro #956B4B, foram selecionadas para proporcionar contraste e equilíbrio visual à interface. Calibri é uma fonte sans-serif moderna e amplamente utilizada, conhecida por sua legibilidade em tamanhos pequenos. Isso a torna uma escolha excelente para textos longos, como parágrafos e blocos de conteúdo, onde a clareza é essencial para a compreensão do usuário, além de possuir uma forma limpa e

simplificada, o que a torna acessível para uma ampla variedade de públicos, incluindo aqueles com dificuldades de visão.

Todos os objetos dispostos na tela visam respeitar a hierarquia visual que desempenha um papel fundamental na orientação dos usuários através da interface.

1. O contraste de cores é uma ferramenta que se destaca do restante da interface por meio de diferenças de cor, tamanho, forma ou estilo atraem mais atenção dos usuários. O uso de contraste ajuda a destacar elementos importantes, como títulos, botões de ação ou mensagens de erro, tornando-os mais facilmente identificáveis e acessíveis, como podemos ver na figura 17:

Figura 17 - Protótipo tela



Fonte: Elaborado pela autora (2024)

- a. Botão Adicionar livro e Configuração vão ser de uma cor diferente dos outros, assim constatando que eles são os mais importantes.
- b. O tamanho dos elementos na interface é um indicador importante de sua importância e relevância. Elementos maiores tendem a atrair mais atenção do que elementos menores.
- c. Lista inicial de livros no centro da aplicação
- d. O espaçamento adequado entre elementos ajuda a organizar a interface e a criar uma sensação de clareza e ordem. O uso de espaçamento consistente e proporcional melhora a legibilidade e a compreensão da informação.
- e. O alinhamento consistente de elementos cria uma sensação de ordem e coesão na interface. Elementos alinhados verticalmente ou horizontalmente parecem relacionados entre si e são mais fáceis de processar visualmente.

f. A organização visual refere-se à disposição dos elementos na interface de uma forma que facilite a compreensão e a interação dos usuários. Elementos relacionados devem ser agrupados logicamente e apresentados de uma maneira que siga padrões de uso comuns.

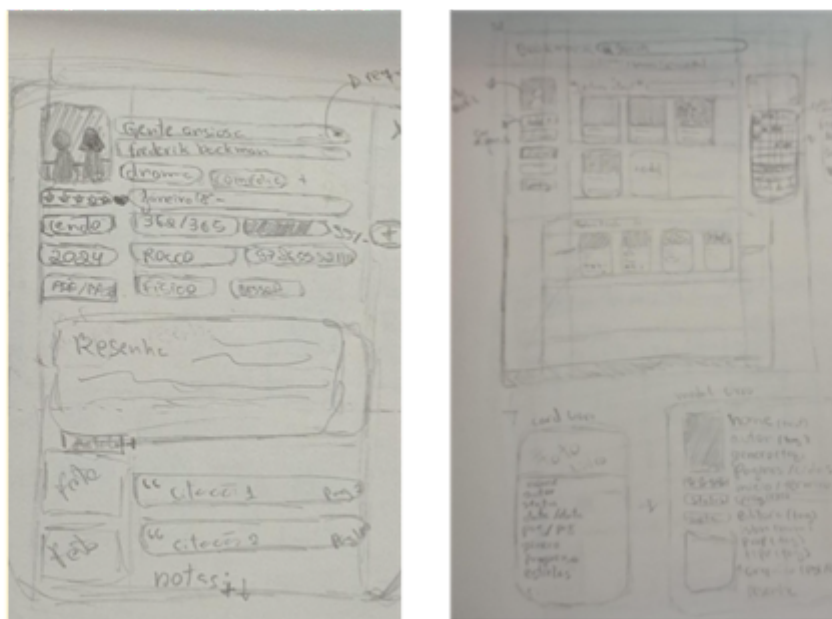
Os botões foram idealizados com *drop-shadow* e cores de destaque para garantir consistência no layout e criar uma hierarquia visual clara. O contraste, o tamanho, o espaçamento e o alinhamento dos elementos foram cuidadosamente considerados para facilitar a compreensão e a interação dos usuários com a interface, que com a possibilidade de mudança de cores e mudanças de tamanho de fontes podem vir a se tornar acessíveis com as alterações necessárias.

Além disso, o menu lateral foi posicionado à esquerda da interface, seguindo padrões de leitura ocidentais, afinal colocar o menu no lado esquerdo da interface segue esse padrão natural de leitura, permitindo que os usuários comecem sua jornada de navegação a partir do canto superior esquerdo, o ponto de início tradicional para a leitura e garantindo familiaridade, acessibilidade e hierarquia visual na navegação da aplicação.

5.4 Construção do Protótipo

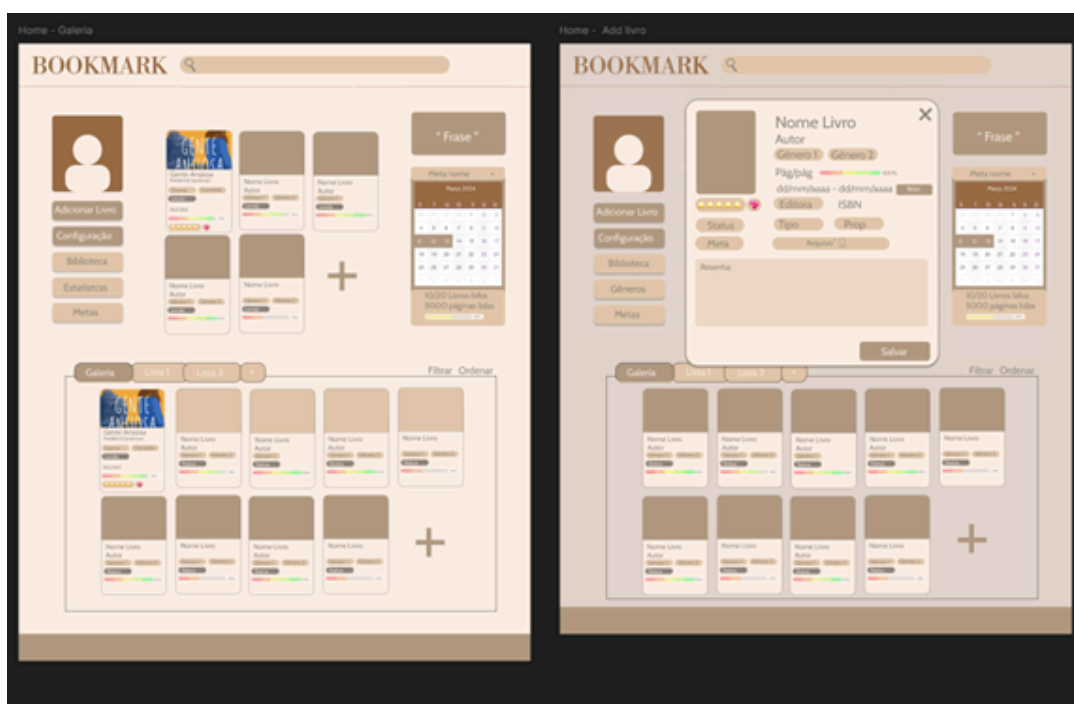
A criação dos protótipos foi conduzida em diferentes níveis de fidelidade, desde rascunhos de baixa fidelidade, desenhados em caderno à mão livre, até protótipos interativos de alta fidelidade. Esses protótipos de alta fidelidade foram desenvolvidos utilizando ferramentas de design e prototipação adequadas, como o Figma, uma ferramenta de design de interface e prototipagem, seguindo os elementos de design definidos.

Nas imagens a seguir são apresentadas as prototipagens de telas de baixa e alta fidelidade. A figura 18 representa os primeiros protótipos de baixa fidelidade da aplicação BookMark. São uma representação rápida e simplificada, utilizadas para explorar ideias sobre composição e fluxo.

Figura 18 - Rascunho de baixa fidelidade

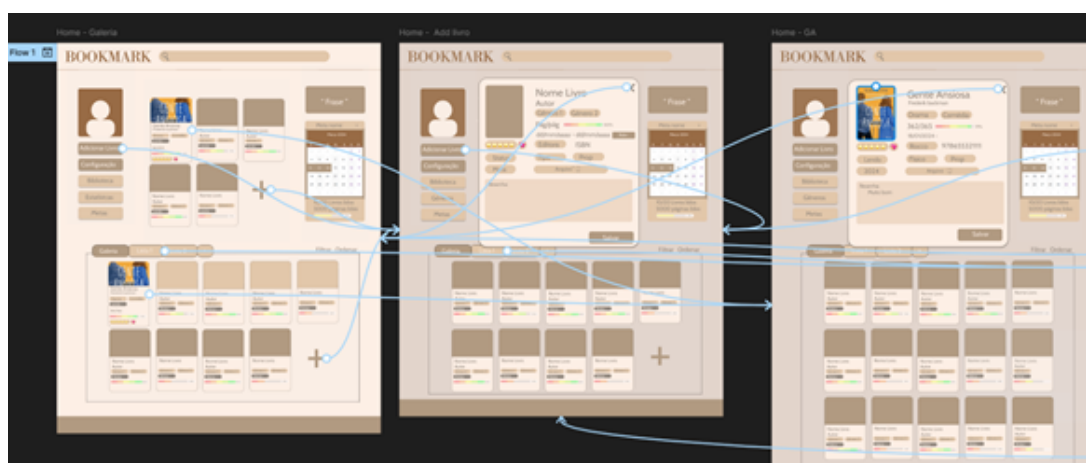
Fonte: Elaborado pela autora (2024)

A Figura 19, ao apresentar uma visão geral dos modelos desenvolvidos no Figma, fornece um ponto de partida essencial para o desenvolvimento do *front-end* com React. Cada tela prototipada e seus componentes correspondentes podem ser visualizados de forma abrangente, permitindo que os desenvolvedores compreendam a estrutura e a organização da interface do usuário. Ao relacionar isso com o desenvolvimento em React, os componentes prototipados podem ser facilmente traduzidos em pequenos componentes, facilitando a implementação da lógica de visualização e interação. Isso ajuda a garantir consistência e eficiência no desenvolvimento do *front-end*, permitindo que a equipe se concentre na criação de uma experiência de usuário coesa e intuitiva.

Figura 19 - Prototipação de Alta fidelidade

Fonte: Elaborado pela autora (2024)

A Figura 20 fornece uma visão do fluxo geral entre as telas modeladas no Figma, destacando a sequência de interações do usuário dentro da interface. Isso ajuda os desenvolvedores a entenderem como as diferentes telas estão conectadas e como os usuários irão navegar pela aplicação. Essa visão do fluxo geral no Figma pode ser utilizada como referência durante a implementação do *front-end*, garantindo que a navegação entre as telas seja fluida e intuitiva para os usuários.

Figura 20 - Protótipo de Alta fidelidade – Fluxo

Fonte: Elaborado pela autora (2024)

A Figura 21 apresenta a tela inicial da aplicação, fornece uma visão detalhada dos elementos e funcionalidades que serão implementados no *front-end* utilizando React. A implementação desses elementos em React é facilitada pelo uso de componentes reutilizáveis.

Figura 21 - Protótipo de Alta fidelidade - Tela Inicial



Fonte: Elaborado pela autora (2024)

Os *cards* dos livros, que exibirão as informações dos livros adicionados pelo usuário, podem ser representados por um componente BookCard, que recebe os dados do livro como *props* e renderiza as informações correspondentes. O calendário com metas de leituras e as listas de leitura do usuário também podem ser implementados como componentes separados, oferecendo modularidade e reusabilidade ao código.

Esse protótipo fornece uma representação visual e interativa das funcionalidades da aplicação antes mesmo de começar a escrever o código. O protótipo da figura 22 representa o de cadastro de livro, uma funcionalidade essencial da aplicação. Esse cadastro pode ser desenvolvido como um componente reutilizável, que pode ser renderizado e acionado em diferentes partes da aplicação sempre que necessário.

Figura 22- Protótipo de Alta fidelidade - Cadastro Livro

The figure displays three high-fidelity prototypes of a book registration modal. Each modal has a title bar with a close button (X). The first modal is a blank form with the following fields: 'Nome Livro' (text input), 'Autor' (text input), 'Gênero 1' and 'Gênero 2' (selects), 'Pág/pág' (range input), 'Editora' (text input), 'ISBN' (text input), 'Status' (select), 'Tipo' (select), 'Preço' (text input), and a 'Resenha' (text area). The second modal shows the 'Photo' field highlighted with a blue border. The third modal shows the form filled with data for the book 'Gente Ansiosa' by Frederik backman, including a cover image, genres (Drama, Comédia), rating (362/365), publication date (18/01/2014), ISBN (9786553211111), and status (Lendo). A 'Salvar' button is at the bottom of each modal.

Fonte: Elaborado pela autora (2024)

O componente de Cadastro de Livro pode conter campos para inserção das informações do livro, como capa, nome, autor, entre outros. Esses campos podem ser representados por componentes individuais, como *inputs* ou *selects*, e organizados dentro do modal de acordo com o *layout* definido no protótipo.

Além disso, o componente pode incluir botões de ação para confirmar o cadastro do livro ou fechar o *modal*. Esses botões foram implementados como componentes *Button*, que podem ser estilizados conforme as diretrizes de design da aplicação. A final interação do usuário com o modal, como preenchimento dos campos e acionamento dos botões, pode ser controlada por estados internos do componente *modal*, utilizando o *useState* do React para gerenciar o estado dos campos de entrada e o *useEffect* para lidar com ações assíncronas, como o envio dos dados do livro para o servidor.

Na implementação em React, o modal de configurações representado pela Figura 23 pode ser desenvolvido como um componente separado, responsável por exibir opções de personalização da interface para o usuário. Esse componente pode ser acionado por meio de um botão de configurações no menu.

Figura 23 - Protótipo de Alta fidelidade - Tela Configuração



Fonte: Elaborado pela autora (2024)

Dentro do modal de configurações, os usuários terão a possibilidade de ajustar diferentes aspectos visuais da aplicação, como as cores de fundo, as fontes utilizadas e seus tamanhos. Essas opções podem ser apresentadas por meio de campos de seleção, sliders ou outros elementos de interface adequados para cada tipo de configuração.

Além disso, é importante considerar a persistência das configurações do usuário, permitindo que elas sejam salvas localmente no navegador ou armazenadas no servidor, conforme a necessidade da aplicação.

5.5 Implementação, entrega e *feedback*

5.5.1 Implementação

A implementação do projeto foi realizada com a biblioteca React para desenvolvimento *web*. O usuário tem a possibilidade de cadastrar livros, editar e atualizar suas leituras, além de definir e acompanhar metas de leitura. Além disso, o layout pode ser personalizado, permitindo que o usuário ajuste as configurações visuais de acordo com suas preferências. O projeto está disponível para acesso em: <https://lpxz8z-3000.csb.app/>.

Este projeto se concentra no desenvolvimento do *front-end* de uma aplicação *web* para gerenciar atividades de leitura. Portanto, não será abordado em detalhes a implementação do banco de dados, que faz parte de um projeto correlato. Contudo, como se trata de uma aplicação *web*, é possível afirmar que todos os dados do *front-end* serão armazenados como objetos JSON em um banco de dados NoSQL hospedado na nuvem, utilizando o Firebase (Firebase, 2024).

5.5.2 Código React

O desenvolvimento da aplicação utilizou o React como tecnologia principal para a construção da interface, proporcionando uma interação dinâmica e fluida para o usuário. A arquitetura do sistema foi planejada para gerenciar estados locais e globais, além de integrar funcionalidades avançadas como persistência de dados no *localStorage* e manipulação do DOM.

Os estados são gerenciados principalmente com o *hook useState*, permitindo que dados, como a lista de livros, metas e configurações, sejam atualizados em tempo real. A atualização do DOM e outras interações específicas são gerenciadas com o *useEffect*, que sincroniza estados e aplica personalizações na interface, como alterações nas cores da aplicação.

O sistema conta com funções específicas para adicionar e gerenciar livros, como a função *addBook*, que permite inserir novos livros na biblioteca. Ela é projetada para atualizar o estado global, salvar os dados no *localStorage* e ajustar elementos adicionais, como destacar datas no calendário e atualizar o progresso de metas de leitura. Para garantir que o progresso das metas seja corretamente refletido, foi implementada a função *updateGoalProgress*, que calcula a porcentagem de conclusão com base nos livros associados e ajusta automaticamente o *status* da meta para "Concluído" ou "Em andamento".

A renderização da aplicação é feita de forma dinâmica, com base no estado da tela atual. O componente responsável por essa lógica utiliza uma estrutura de controle para determinar qual componente deve ser exibido, como a biblioteca de livros, metas ou gêneros. Isso garante uma experiência de navegação fluida e intuitiva para o usuário.

Além disso, subcomponentes como *Book* e *AddBook* foram criados para gerenciar a exibição e adição de livros. O componente *Book* exibe os detalhes de cada

livro, calculando o progresso de leitura como uma porcentagem e aplicando estilos personalizados à barra de progresso. Já o componente *AddBook* implementa a lógica do formulário para inserir novos livros, utilizando estado para armazenar e atualizar dinamicamente os dados preenchidos pelo usuário.

A aplicação também inclui componentes específicos para a exibição dos livros, como *BookGallery* e *BookList*, que apresentam os dados em diferentes formatos, cartões interativos e tabelas dinâmicas, respectivamente, permitindo uma personalização flexível.

Foram importadas as dependências *useState*, *useRef* e *useEffect* que servem para gerenciar os dados de forma dinâmica no componente principal App.

Na Figura 24, o *useState* é utilizado para gerenciar estados locais. Nesse caso, é inicializado com uma função que recupera os livros armazenados no *localStorage* ou, caso não existam dados salvos, retorna um conjunto de dados fictícios do *mockData*.

Figura 24 – *useState*

```
17 |   const [books, setBooks] = useState(() => {  
18 |     const savedBooks = JSON.parse(localStorage.getItem("books"));  
19 |     return savedBooks || mockData.books;  
20 |   });  
--
```

Fonte: Elaborado pela autora (2024)

Na Figura 25, o *useEffect* é responsável por sincronizar os estados com os dados armazenados no *localStorage*, além de aplicar configurações personalizadas ao DOM, como a definição de cores da aplicação. Ele também carrega metas salvas anteriormente, garantindo que a interface seja carregada com os valores armazenados.

Figura 25 – useEffect

```

68  useEffect(() => {
69      const savedGoals = JSON.parse(localStorage.getItem("goals")) || [];
70      setGoals(savedGoals);
71
72      const savedCorFundoPrincipal = localStorage.getItem("corFundoPrincipal") || "#F0F0F0";
73      const savedCorTextoPrincipal = localStorage.getItem("corTextoPrincipal") || "#986B45";
74      const savedCorFundoSecundaria = localStorage.getItem("corFundoSecundaria") || "#E2C7AA";
75      const savedCorContraste = localStorage.getItem("corContraste") || "#A0522D";
76      const savedCorTerciaria = localStorage.getItem("corTerciaria") || "#B19A81";
77      const savedCorFonteClara = localStorage.getItem("corFonteClara") || "#FFFFFF";
78      const savedCorFonteEscura = localStorage.getItem("corFonteEscura") || "#333333";
79
80      setCorFundoPrincipal(savedCorFundoPrincipal);
81      setCorTextoPrincipal(savedCorTextoPrincipal);
82      setCorFundoSecundaria(savedCorFundoSecundaria);
83      setCorContraste(savedCorContraste);
84      setCorTerciaria(savedCorTerciaria);
85      setCorFonteClara(savedCorFonteClara);
86      setCorFonteEscura(savedCorFonteEscura);
87
88      document.documentElement.style.setProperty("--cor-fundo-principal", savedCorFundoPrincipal);
89      document.documentElement.style.setProperty("--cor-texto-principal", savedCorTextoPrincipal);
90      document.documentElement.style.setProperty("--cor-fundo-secundaria", savedCorFundoSecundaria);
91      document.documentElement.style.setProperty("--cor-contraste", savedCorContraste);
92      document.documentElement.style.setProperty("--cor-terciaria", savedCorTerciaria);
93      document.documentElement.style.setProperty("--cor-fonte-clara", savedCorFonteClara);
94      document.documentElement.style.setProperty("--cor-fonte-escura", savedCorFonteEscura);
95  }, []);

```

Fonte: Elaborado pela autora (2024)

Na Figura 26, o código apresentado implementa a função *addBook*, cujo objetivo é adicionar um novo livro à lista de livros. Essa função recebe como parâmetro o objeto *newBook*, que contém as informações do livro, e após isso, o estado é atualizado e os dados são salvos no *localStorage*.

Para atualizar o estado, é criado um *array* utilizando o operador *spread* *...prevBooks*, que combina o estado anterior *prevBooks* com o novo livro. Em seguida, o código verifica se o livro possui datas de início e término. Caso possua, os dias correspondentes a essas datas são destacados no calendário da aplicação. Por fim, o código avalia se o livro possui uma meta associada.

Figura 26 - Adicionar e Gerenciar Livros

```

114   const addBook = (newBook) => {
115       setBooks((prevBooks) => {
116           const updatedBooks = [...prevBooks, newBook];
117           saveBooksToLocalStorage(updatedBooks);
118           return updatedBooks;
119       });
120
121       if (newBook.startDate && newBook.endDate) {
122           updateHighlightedDays(newBook.startDate, newBook.endDate);
123       }
124
125       if (newBook.goal) {
126           updateGoalProgress(newBook.goal);
127       }
128   };

```

Fonte: Elaborado pela autora (2024)

Na figura 27 o código define a função *renderContent*, responsável por renderizar componentes dinâmicos da aplicação com base no estado atual da tela *currentScreen*. Essa lógica é implementada utilizando uma estrutura de *switch case*, que avalia o valor de *currentScreen* para determinar qual conteúdo deve ser exibido.

No caso em que a tela atual é "biblioteca", a função retorna o componente *LibraryScreen*, passando os livros, a busca, a função para controlar a abertura do modal *setIsModalOpen*, e as funções para adicionar, atualizar, deletar livros e metas. Quando a tela atual é "metas", o componente retornado é *GoalPage*, que recebe os livros, as metas, e uma função para adicionar novas metas. Se a tela atual for "genres", a função retorna o componente *Genre*, que recebe apenas os livros como propriedade

No caso padrão, quando nenhuma das telas anteriores corresponde ao estado atual, o código renderiza um componente *Library*. Ele recebe as mesmas propriedades do *LibraryScreen*, afinal é uma alternativa para exibir a biblioteca.

Além disso, um botão é exibido nesta tela, permitindo a abertura de um *modal* para adicionar livros.

Figura 27 – Renderização

```

236   const renderContent = () => {
237     switch (currentScreen) {
238       case "biblioteca":
239         return (
240           <LibraryScreen
241             books={books}
242             searchQuery={searchQuery}
243             setIsModalOpen={setIsModalOpen}
244             addNewBook={addBook}
245             onUpdateBook={updateBook}
246             onDeleteBook={deleteBook}
247             goals={goals}
248           />
249         );
250       case "metas":
251         return <GoalPage books={books} goals={goals} addGoal={addGoal} />;
252       case "genres":
253         return <Genre books={books} />;
254       default:
255         return (
256           <div>
257             <Library

```

Fonte: Elaborado pela autora (2024)

Na figura 28, podemos ver um padrão de desestruturação do livro passado, para que caso alguma dessas informações listadas chegarem vazias serão atribuídos esses valores padrões.

Figura 28 - Dados em Book

```

8     const {
9       currentPage = 0,
10      totalPages = 0,
11      name = 'Nome desconhecido',
12      author = 'Autor desconhecido',
13      genre = 'Gênero não especificado',
14      status = 'Status desconhecido',
15      image,
16      rating = 'Sem avaliação'
17    } = book;

```

Fonte: Elaborado pela autora (2024)

Na figura 29, o progresso é calculado como porcentagem com base na *currentPage* e no *totalPages*. Além da função auxiliar que define a cor da barra de progresso da porcentagem de páginas lidas no livro.

Figura 29 - Progresso em Book

```

19  const progress = totalPages > 0 ? (currentPage / totalPages) * 100 : 0;
20
21  const getProgressBarColor = (progress) => {
22    if (progress < 20) return '#FF4C4C';
23    if (progress < 40) return '#FF8C4C';
24    if (progress < 60) return '#FFD74C';
25    if (progress < 80) return '#B2FF4C';
26    return '#4CFF4C';
27  };
28

```

Fonte: Elaborado pela autora (2024)

O componente terciário *BookGallery*, na figura 30, exibe uma galeria de livros em formato de lista, onde cada livro é apresentado como um cartão interativo. Ele suporta exibição de campos dinâmicos baseados nas opções fornecidas.

Figura 30 – BookGallery

```

5  <div className="book-list">
6    {books.map((book) => {
7      const bookGoal = goals.find((goal) => String(goal.id) === String(book.goal));
8      return (
9        <div key={book.id} className="book-card" onClick={() => onBookClick(book)}>
10         <img src={book.image} alt={`Capa de ${book.name}`} className="book-image" />
11         {fields.includes('nome') && <h3>{book.name}</h3>}
12         {fields.includes('autor') && <p>Autor: {book.author}</p>}
13         {fields.includes('genero') && <p>Gênero: {book.genre}</p>}
14         {fields.includes('totalPaginas') && <p>Páginas Totais: {book.totalPages}</p>}
15         {fields.includes('paginaAtual') && <p>Páginas Atuais: {book.currentPage}</p>}
16         {fields.includes('editora') && <p>Editora: {book.publisher}</p>}
17         {fields.includes('isbn') && <p>ISBN: {book.isbn}</p>}
18         {fields.includes('tipo') && <p>Tipo: {book.type}</p>}
19         {fields.includes('propriedade') && <p>Propriedade: {book.prop}</p>}
20         {fields.includes('arquivo') && (
21           <p>
22             Arquivo: {book.file ? <a href={book.file} download>Baixar</a> : 'Nenhum arquivo disponível'}
23           </p>
24         )}
25       </div>
26     )}
27   </div>

```

Fonte: Elaborado pela autora (2024)

O componente *BookList*, na figura 31, exibe uma lista de livros em formato de tabela tr, com campos dinâmicos definidos por meio de uma *prop fields*, de forma semelhante a *BookGallery*.

Figura 31 – BookList

```

43 <tr key={book.id} onClick={() => onBookClick(book)}>
44   {fields.includes('imagem') && (
45     <td>
46       {book.image ? (
47         <img src={book.image} alt={`Capa do Livro: ${book.name} || 'Sem Título'`} className="book-image" />
48       ) : (
49         <span>Sem Imagem</span>
50       )}
51     </td>
52   )}
53   {fields.includes('nome') && <td>{book.name}</td>}
54   {fields.includes('autor') && <td>{book.author}</td>}
55   {fields.includes('genero') && <td>{book.genre || "Gênero não especificado"}</td>}
56   {fields.includes('totalPaginas') && <td>{book.totalPages}</td>}
57   {fields.includes('paginaAtual') && <td>{book.currentPage}</td>}
58   {fields.includes('editora') && <td>{book.publisher || "Editora não especificada"}</td>}
59   {fields.includes('isbn') && <td>{book.isbn}</td>}
60   {fields.includes('tipo') && <td>{book.type || "Tipo não especificado"}</td>}
61   {fields.includes('propriedade') && <td>{book.prop || "Propriedade não especificada"}</td>}
62   {fields.includes('arquivo') && (
63     <td>
64       {book.file ? (
65         <a href={book.file} download>Baixar</a>
66       ) : (
67         <span>Nenhum arquivo disponível</span>
68       )}
69     </td>
70   )}
71   {fields.includes('avaliacao') && <td>{book.rating}</td>}
72   {fields.includes('status') && <td>{book.status}</td>}
73   {fields.includes('meta') && <td>{bookGoal ? bookGoal.name : "Sem meta definida"}</td>}
74   {fields.includes('resenha') && <td>{book.review}</td>}
75   {fields.includes('dataInicio') && <td>{book.startDate}</td>}
76   {fields.includes('dataTermino') && <td>{book.endDate}</td>}
77 </tr>

```

Fonte: Elaborado pela autora (2024)

A função *saveGoal*, demonstrada na figura 32, é responsável por salvar e atualizar metas na aplicação, garantindo a validação dos campos obrigatórios, a atualização do estado das metas e o *localStorage*.

Começa verificando se os campos obrigatórios da nova meta foram preenchidos. Os campos são sinalizados com um asterisco, para mostrar que são obrigatórios, caso um desses campos esteja vazio, é exibido um alerta para o usuário com a mensagem: "Por favor, preencha todos os campos obrigatórios.", e a execução da função é interrompida com *return*. Se os campos obrigatórios estiverem preenchidos, a função prossegue criando ou atualizando a lista de metas. Para isso, verifica se a meta que está sendo editada ou se já existe. Caso exista, utiliza o método *map* para percorrer todas as metas e encontra aquela que corresponde ao *id* da meta sendo editada. A meta encontrada é atualizada com os novos dados.

No caso da criação de uma nova meta, um novo objeto de meta é adicionado ao array *goals* utilizando o operador *spread* para preservar as metas existentes e incluindo o novo objeto de meta, que recebe um *id* gerado com a função *Date.now()* para garantir unicidade, além das propriedades de *newGoal*.

Após a lista de metas ser criada ou atualizada, o estado global das metas é atualizado com a função *setGoals(updatedGoals)*. Em seguida, os dados atualizados são salvos no *localStorage* por meio da função *saveGoalsToLocalStorage(updatedGoals)*.

Figura 32 - Salvando meta

```

46 | const saveGoal = () => {
47 |   if (!newGoal.name || !newGoal.totalBooks) {
48 |     alert('Por favor, preencha todos os campos obrigatórios.');
```

```

49 |     return;
50 |   }
51 |
52 |   let updatedGoals;
53 |   if (editingGoal) {
54 |     updatedGoals = goals.map((goal) =>
55 |       goal.id === editingGoal.id ? { ...goal, ...newGoal } : goal
56 |     );
57 |   } else {
58 |     updatedGoals = [...goals, { id: Date.now(), ...newGoal }];
59 |   }
60 |
61 |   setGoals(updatedGoals);
62 |   saveGoalsToLocalStorage(updatedGoals);
63 |   closeModal();
64 | };

```

Fonte: Elaborado pela autora (2024)

5.5.3 Aplicação

No projeto, o usuário tem a opção de adicionar um novo livro à sua lista de leitura de maneira prática e intuitiva. Esse processo pode ser iniciado por meio de dois botões: o botão “Adicionar livro”, que se encontra no menu lateral, ou o botão “+” na tela inicial. Ao clicar em qualquer um desses botões, o sistema abre um modal específico para o cadastro de livros, ao cadastrar um livro o livro é adicionado na tela inicial, como pode ser visto na figura 33.

Nesta janela, o usuário pode inserir diversas informações sobre o livro, como título, autor, gênero, número de páginas, *status* de leitura, capa, entre outros. Dentre esses dados, o único campo obrigatório é o nome do livro, garantindo que o cadastro seja simples e acessível. Após preencher as informações desejadas, o usuário clica no botão "Salvar". Em seguida, o livro é automaticamente adicionado à tela inicial da aplicação e aparecendo na lista de livros do usuário.

Figura 33 - Cadastro de livro

The screenshot shows the 'BOOKMARK' application interface. A modal window is open for adding a new book. The modal contains the following fields and elements:

- Capa:** A small image placeholder for the book cover.
- Título:** A text input field containing 'Gente ansiosa'.
- Autor:** A text input field containing 'Fredrik Backman'.
- ISBN:** Two input fields containing '421' and '133'.
- Data Início:** A date picker showing '13/11/2024'.
- Data Término:** A date picker showing 'dd/mm/aaaa'.
- Lendo:** A dropdown menu showing 'Me'.
- Arquivo:** A section with radio buttons for 'Ficção', 'Físico', 'Pessoal', and 'Arquivo'.
- Progresso:** A progress bar showing '100% concluído'.
- Calendar:** A calendar for November 2024.
- Buttons:** 'Salvar' (Save) and a plus sign (+) to add more books.

Fonte: Elaborado pela autora (2024)

Na Figura 34, demonstra-se a interação do usuário com a interface de gerenciamento de livros. Ao clicar em um *card* de livro, que se assemelha a um marcador de página (Figura 34 a), previamente cadastrado, um novo *modal* é exibido (Figura 34 b). Esse *modal* traz todas as informações já registradas do livro, permitindo ao usuário visualizar e editar esses dados com facilidade (Figura 34 c).

Figura 34 - Atualização do livro

Fonte: Elaborado pela autora (2024)

Na Figura 35, podemos ver a funcionalidade de personalização de cores e tamanhos de fontes da página, acessível pelo botão configurações. Esse recurso permite que o usuário ajuste a paleta de cores e o tamanho das fontes conforme suas preferências, promovendo não apenas uma experiência visual mais agradável, mas também um aumento significativo na acessibilidade.

Ao proporcionar a liberdade de alterar cores e fontes quantas vezes for necessário, essa funcionalidade atende a necessidades específicas, como facilitar a visualização para pessoas com baixa visão, que podem se beneficiar com fontes maiores, e tornar o site mais acessível para daltônicos, que podem ajustar as cores para melhor contraste e diferenciação dos elementos visuais. Essa flexibilidade reforça o objetivo de uma interface inclusiva e adaptável, que atende a uma variedade de perfis de usuários.

Figura 35 - Configurações do layout

The image shows a settings panel with two main sections: 'Alterar Fontes' and 'Alterar Cores do Tema'. The 'Alterar Fontes' section on the left contains four input fields for font sizes: 'Tamanho da Fonte Pequena' (12), 'Tamanho da Fonte Normal' (16), 'Tamanho da Fonte Média' (20), and 'Tamanho da Fonte Grande' (24). Below these are two color selection boxes for 'Cor da Fonte Clara' (black) and 'Cor da Fonte Escura' (white). The 'Alterar Cores do Tema' section on the right contains five color selection boxes: 'Cor de Fundo Principal' (light blue), 'Cor do Texto Principal' (dark blue), 'Cor de Fundo Secundária' (light blue), 'Cor de Contraste Principal' (dark blue), and 'Cor Terciária' (medium blue). A brown 'Aplicar' button is located between the two sections. A close button (X) is in the top right corner of the panel.

Fonte: Elaborado pela autora (2024)

A Figura 36 apresenta a tela de cadastro de metas, uma funcionalidade essencial para o gerenciamento das metas de leitura do usuário. Com essa ferramenta, o usuário pode criar metas personalizadas, inserindo informações como uma capa de sua escolha, nome da meta, quantidade de livros (campo obrigatório), data de início, data de término e status da meta.

Figura 36 - Cadastro de metas

O formulário de cadastro de metas é exibido em uma interface web. No topo, há uma barra de busca com o texto "Pesquisar...". Abaixo, há uma barra de navegação com os links "Início", "Gêneros" e "Metas". O formulário principal é dividido em duas colunas. À esquerda, há uma seção "Capa" com um ícone de capa de livro e uma imagem de uma paisagem com pedras e plantas. À direita, há campos para "Nome da Meta" (com o valor "Novembro"), "Total de Livros" (com o valor "2"), "Data de Início" (com o valor "07/11/2024" e um ícone de calendário), "Data de Término" (com o valor "dd/mm/aaaa" e um ícone de calendário) e "Status" (com o valor "Em andamento"). Um botão "Salvar" está localizado na parte inferior direita do formulário.

Fonte: Elaborado pela autora (2024)

Na Figura 37, observamos o processo de atualização automática de uma meta de leitura. Esse recurso permite que o progresso da meta seja recalculado de forma dinâmica sempre que um livro associado a ela atinge 100% de progresso, ou seja, quando a página atual do livro corresponde à sua página total.

Quando isso ocorre, a meta atualiza automaticamente o número de livros concluídos, refletindo o progresso real do usuário. Além disso, essa funcionalidade também exibe uma lista de livros que fazem parte de cada meta, facilitando o acompanhamento visual dos títulos associados.

Figura 37 - Meta atualizada

Metas de Leitura

hoje

Livros a Ler: 0/2

Progresso: 0.00%

0.00%

Data de Início:

Data de Término:

Status: Em andamento

Novembro

Livros a Ler: 1/2

Progresso: 50.00%

50.00%

Data de Início: 2024-11-01

Data de Término:

Status: Em andamento

Livros nesta Meta:

Os dois livros ao final

Modal Form:

Capa

Nome da Meta: Novembro

Total de Livros: 2

Data de Início: 07/11/2024

Data de Término: dd/mm/aaaa

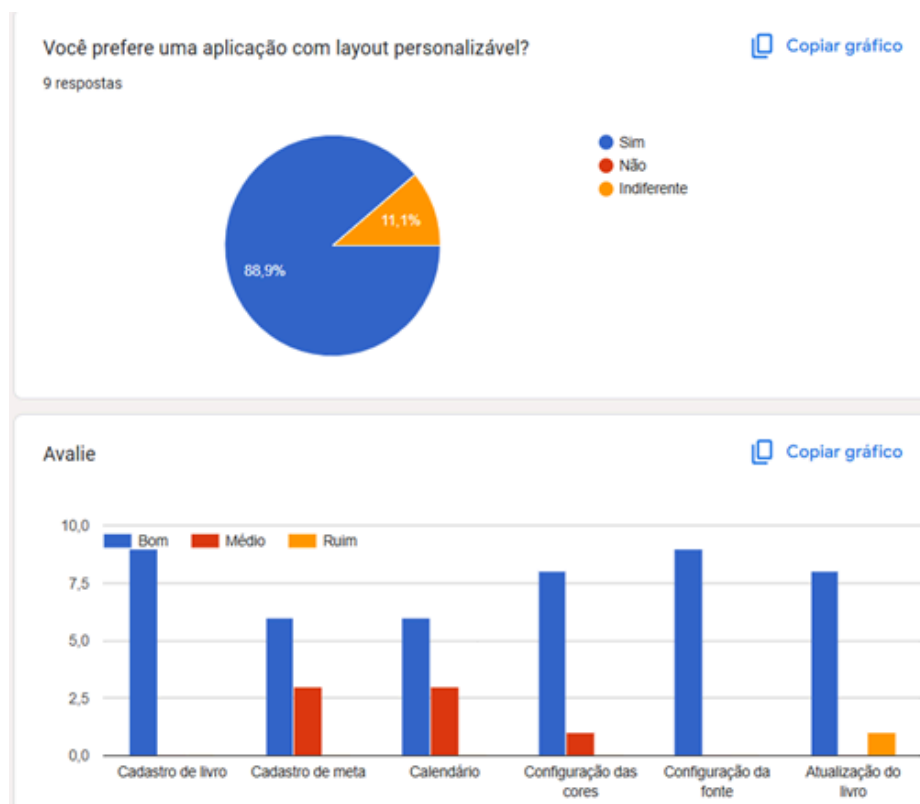
Status: Status

Salvar

Fonte: Elaborado pela autora (2024)

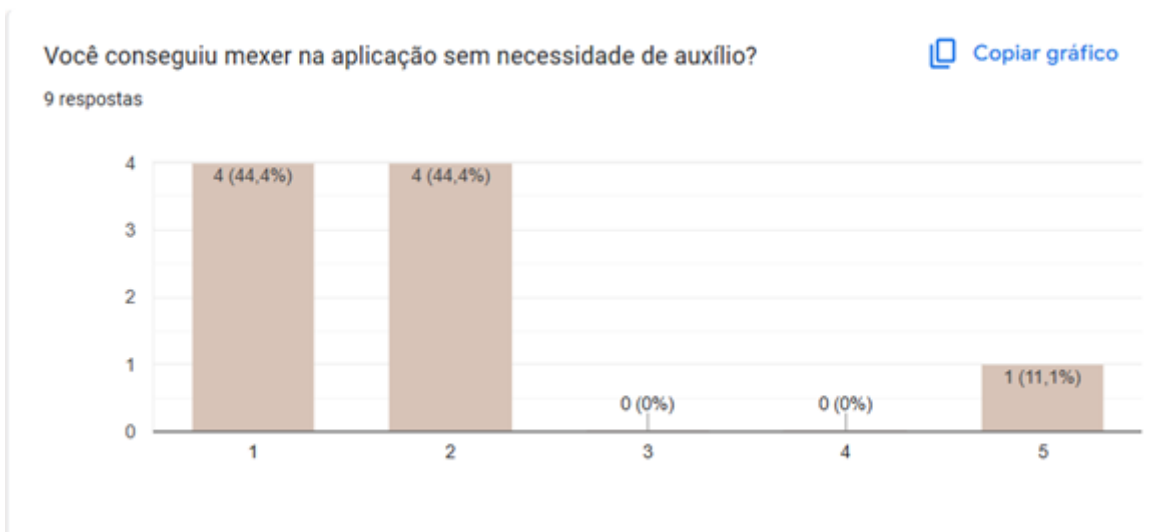
5.5.4 Entrega e Feedback

A forma de avaliação dos resultados e a aplicação dos testes de usabilidade seguiram um processo estruturado que garantiu que a aplicação atendesse às necessidades dos usuários finais. Os testes de usabilidade foram conduzidos em várias etapas do desenvolvimento, começando com testes funcionais e de integração para assegurar que todos os componentes da aplicação estão operando corretamente. Em seguida, foram aplicados testes de usabilidade com um grupo selecionado de usuários finais para avaliar a interface e a experiência do usuário, foram observados aspectos como facilidade de uso, eficiência e satisfação, como pode ser visto na figura 38.

Figura 38 – Questionário

Fonte: Elaborado pela autora (2024)

Para avaliar a funcionalidade do sistema, realizamos testes com um grupo de usuários selecionados. Esses usuários foram responsáveis por testar as funções de personalização do layout. Durante os testes, foi observada a intuitividade do sistema, mensurada pela necessidade (ou falta) de assistência requerida pelos usuários ao realizar as personalizações, como pode ser vista na figura 39 onde 1 significa que não foi necessário auxílio e 5 que foi necessário auxílio.

Figura 39 – Questionário

Fonte: Elaborado pela autora (2024)

O feedback final foi coletado por meio de questionário. Este feedback foi analisado e utilizado para fazer ajustes e melhorias na aplicação. Com base nisso foram feitas as modificações necessárias e integrações sobre as funcionalidades e a interface, garantindo uma evolução contínua da aplicação.

6 CONSIDERAÇÕES FINAIS

A aplicação desenvolvida atendeu aos objetivos propostos, fornecendo uma solução prática e eficiente para o gerenciamento de leitura pessoal.

Um dos objetivos era comparar as interfaces de aplicações existentes e entender seus problemas, focando na experiência do usuário, funcionalidades e acessibilidades. Foram analisadas as aplicações Skoob, Goodreads e Maratona. As funcionalidades dessas aplicações inspiraram as funcionalidades do BookMark e as limitações observadas foram superadas com uma interface mais moderna e centrada nas necessidades individuais dos leitores.

Quanto às funcionalidades, o objetivo era desenvolver uma aplicação que permitisse aos leitores organizarem suas leituras. Este objetivo foi atingido por meio da criação de um sistema de metas de leitura.

Com relação ao objetivo de desenvolver o design de interface focado em UX/UI, garantindo uma experiência intuitiva, agradável e acessível para os usuários, buscou-se a seleção de tons de cores suaves e agradáveis, que oferecesse uma experiência visualmente atraente. Além disso, a combinação de cores foi projetada levando em conta a psicologia das cores, visando criar uma atmosfera relaxante e acolhedora que incentive os usuários a explorarem e interagirem com o conteúdo da aplicação. Em contraste com a abordagem tradicional de se utilizar predominantemente preto e branco em interfaces de aplicações, a paleta de cores proposta oferece uma alternativa mais dinâmica e envolvente.

Do ponto de vista técnico, adotou-se o framework React visando empregar a tecnologia mais recente e popular em desenvolvimento *web*, complementando assim a formação técnica da autora. O uso do react permitiu a organização do código em componentes, aumentando o reuso dos mesmos, simplificando a arquitetura do site e melhorando a produtividade no desenvolvimento dele. A abordagem de desenvolvimento baseada em prototipação permitiu a aplicação dos conhecimentos de engenharia de software aprendidos ao longo do curso.

Além de todos os objetivos atingidos, o projeto contribui para a cultura leitora, promovendo a organização de leitura em uma plataforma inclusiva e adaptável às preferências de cada usuário.

REFERÊNCIAS

ALURA. **O que é Front-end Back-end e full stack** - aprenda as diferenças entre essas áreas. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>. Acesso em: 11 de jun. 2024

BEZERRA, David de Jesus; GANDELMAN, Isabela Baraldi. **Livronauta gerenciador de leituras**. 2023. Trabalho de Conclusão de Curso (Superior de Tecnologia em Sistemas para Internet) - Faculdade de Tecnologia Rubens Lara, 2023 Disponível em: <https://ric.cps.sp.gov.br/bitstream/123456789/14482/1/LivroNauta.pdf> . Acesso em: 17 mar. 2024.

BRITO, Danielle Santos de. **A importância da leitura na formação social do indivíduo**. 2010 - ISSN 1982-646X. Disponível em: http://fals.com.br/novofals/revela/REVELA%20XVII/Artigo4_ed08.pdf .Acesso em: 7 jun. 2024.

CARVALHO, Amanda Maciel de. **A rede social Skoob**: análise da interface sob a perspectiva da Arquitetura da Informação. 2021. Trabalho de Conclusão de Curso (Biblioteconomia) - Universidade Federal do Rio Grande do Sul, Faculdade de Biblioteconomia e Comunicação, Departamento de Ciência da Informação, 2021. Disponível em: <http://hdl.handle.net/10183/230032> . Acesso em: 17 mar. 2024.

COOLORS. **Palette** Disponível em: <https://colors.co/palette/fdefe5-986b4b-e2c7aa-8f857b-b19a81> .Acesso em: 18 mar. 2024

FIREBASE. Disponível em: https://firebase.google.com/?gad_source=1&gclid=Cj0KCQiApNW6BhD5ARIsACmEbKvHj_S2gpmuOBZb2Tr9x9TqgAmVVnZ3d5BQGMjJwDAOtMP5xTxS7TEaAsMEALw_wcB&gclsrc=aw.ds&hl=pt-br . Acesso em: 17 mar. 2024.

GOODREADS. Disponível em: <https://www.goodreads.com/about/us>. Acesso em: 17 mar. 2024.

GUANABARA, Dayane. Plataformas digitais e o hábito de leitura: um estudo sobre a rede Skoob. **COMFILOTEC: Revista da graduação da FAPCOM** 2018. Disponível em: <https://revista.fapcom.edu.br/index.php/revista-comfilotec/article/view/254>. Acesso em: 10 mar. 2024.

HÁBITO de leitura estimula o cérebro e promove benefícios para a saúde mental. **Portal PUCRS**, 29 de outubro de 2024. Disponível em: <https://portal.pucrs.br/blog/habito-de-leitura/>. Acesso em: 12 jun. 2024.

KRUG, Steve. **Não me faça pensar**: uma abordagem de bom senso à usabilidade na web. 2ª ed. Rio de Janeiro: Alta Books, 2014.

L, Andrei. **React descomplicado**: o que é e como começar Já!. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-react-javascript>. Acesso em: 18 mar. 2024.

MOZILLA DEVELOPER NETWORK. **React**: uma biblioteca javascript para a construção de interfaces de usuário. Disponível em:
https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started Acesso em: 21 de março de 2024a.

MOZILLA DEVELOPER NETWORK. **CSS**. Disponível em:
<https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 11 de jun de 2024b.

MOZILLA DEVELOPER NETWORK, **JavaScript**. Disponível em:
<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 11 de jun de 2024c.

MOZILLA DEVELOPER NETWORK. **Visão geral do cliente-servidor**. Disponível em:
https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/First_steps/Client-Server_overview .Acesso em: 21 de março de 2024d.

PECINATO, Beatriz. BookTok, fenômeno das redes sociais, impacta venda e visibilidade de obras literárias. **Jornal da USP**, 19 de dezembro de 2023. Disponível em:
<https://jornal.usp.br/atualidades/booktok-fenomeno-das-redes-sociais-impacta-na-venda-e-visibilidade-de-obras-literarias/>. Acesso em: 17 mar. 2024.

PRESSMAN, Roger S. **Engenharia de software**, 6 ed. Editora McGrawHill: Porto Alegre, 2010. Acesso em: 17 mar. 2024.

RASCIA, Tania. **Getting started with React**. Disponível em:
<https://www.taniarascia.com/getting-started-with-react/> Acesso em: 03 jun. 2024.

REACT. **Crie um novo React App** Disponível em:
<https://pt-br.reactjs.org/docs/create-a-new-react-app.html>. Acesso em: 11 jun. 2024a.

REACT. **Introduzindo JSX**. Disponível em:
<https://pt-br.reactjs.org/docs/introducing-jsx.html>. Acesso em: 11 jun. 2024b.

REACT. **Learn React**. Disponível em: <https://react.dev/learn>. Acesso em: 03 jun. 2024c.

REACT. **ReactDOM**. Disponível em: <https://legacy.reactjs.org/docs/react-dom.html>. Acesso em: 03 jun. 2024d.

REACT. **Thinking in React**. Disponível em: <https://react.dev/learn/thinking-in-react>. Acesso em: 03 jun. 2024e.

REACT. **Uma biblioteca javascript para construir interfaces de usuário**. Disponível em: <https://legacy.reactjs.org/>. Acesso em: 10 de maio de 2024f.

RIBEIRO, Nando. **Design de interface do usuário (UI):** a arte da experiência visual e usabilidade. Disponível em: <https://leomhann.com.br/design-de-interface-do-usuario-ui/> Acesso em: 11 jun. 2024.

SANTOS, Amanda. **Quatro características de uma boa experiência do usuário.** 19 de julho 2021 Disponível em: <https://brasil.uxdesign.cc/quatro-caracteristicas-de-uma-boja-experiencia-do-usuario-eaf-a0ca93187#:~:text=Existem%20quatro%20caracter%C3%ADsticas%20que%20podem,e%20porque%20elas%20s%C3%A3o%20importantes>. Acesso em: 11 jun. 2024

SANTOS, Laura Coelho. **TikTok e livros:** uma análise sobre a influência dos Booktokers no consumo de livros no Brasil. 2024. Trabalho de Conclusão de Curso (Graduação em Comunicação - Habilitação em Publicidade e Propaganda) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2024, p.13. Disponível em: <https://pantheon.ufrj.br/handle/11422/19677> Acesso em: 17 mar. 2024.

SILVA, Ellen Patricia Belfor da; SANTOS, Jhuliano oliveira dos. **A influência do Booktok na formação de novos leitores.** 2023. Trabalho de Conclusão de Curso (Licenciatura em Letras – Português e Inglês) - Instituto Federal de Educação, Ciência e Tecnologia do Amapá, Campus Macapá, 2023. Disponível em: https://sistemas.intercom.org.br/pdf/link_aceite/nacional/11/0816202322155064dd74c61d384.pdf . Acesso em: 17 de mar. 2024

SKOOB. Disponível em: <https://www.skoob.com.br>. Acesso em: 17 mar. 2024.

TOTVS. **Front end:** o que é, como funciona e qual a importância. Disponível em: <https://www.totvs.com/blog/developers/front-end/>. Acesso em: 11 jun. 2024.

VENDAS, Brenda Lima. A influência do Booktok na mudança de hábito de leitura dos jovens durante a pandemia. **Anais de Resumos Expandidos do Seminário Internacional de Pesquisas em Midiatização e Processos Sociais**, [S.l.], v. 1, n. 5, nov. 2022. ISSN 2675-4169. Disponível em: <https://www.midiaticom.org/anais/index.php/seminario-midiatizacao-resumos/article/view/1525>. Acesso em: 17 mar. 2024.

VISUAL STUDIO CODE. **Code faster.** Disponível em: <https://code.visualstudio.com/>. Acesso em 11 jun. 2024.

W3SCHOOLS. **JavaScript HTML DOM.** Disponível em: https://www.w3schools.com/js/js_htmlDOM.asp. Acesso em: 11 jun. 2024a.

W3SCHOOLS. **React tutorial.** Disponível em: <https://www.w3schools.com/REACT/>. Acesso em: 10 mai. 2024b.

W3SCHOOLS. **What is a Front-End developer?.** Disponível em: https://www.w3schools.com/whatis/whatis_frontenddev.asp. Acesso em: 11 jun. 2024c.